



Titre: Processus d'accélération de projets sous contraintes de ressources
Title: avec modes de chevauchement

Auteur: Lucas Greze
Author:

Date: 2011

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Greze, L. (2011). Processus d'accélération de projets sous contraintes de
Citation: ressources avec modes de chevauchement [Master's thesis, École Polytechnique
de Montréal]. PolyPublie. <https://publications.polymtl.ca/596/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/596/>
PolyPublie URL:

**Directeurs de
recherche:** Robert Pellerin, & Patrice Leclaire
Advisors:

Programme: Génie industriel
Program:

UNIVERSITÉ DE MONTRÉAL

PROCESSUS D'ACCÉLÉRATION DE PROJETS SOUS CONTRAINTES
DE RESSOURCES AVEC MODES DE CHEVAUCHEMENT

LUCAS GREZE

DÉPARTEMENT DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTE EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL)

JUIN 2011

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

PROCESSUS D'ACCÉLÉRATION DE PROJETS SOUS CONTRAINTES DE
RESSOURCES AVEC MODES DE CHEVAUCHEMENT

Présenté par : GREZE Lucas

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

M. FORTIN Clément, Ph.D., président

M. PELLERIN Robert, Ph.D., membre et directeur de recherche

M. LECLAIRE Patrice, Doct., membre et codirecteur de recherche

M. HAJJI Adnène, Ph. D., membre

REMERCIEMENTS

Je voudrais remercier avant tout mes directeurs de recherche Messieurs Robert Pellerin et Patrice Leclaire, qui m'ont offert un encadrement et un support d'une grande qualité tout au long de ma recherche. Leurs conseils m'ont été d'une aide précieuse et leur disponibilité fort appréciable.

Je ne saurais jamais assez remercier François Berthaut et Nathalie Perrier pour l'aide importante qu'ils m'ont offert. Leurs expériences et leurs disponibilités m'ont été extrêmement profitable pour mener à bien mon projet de recherche.

J'adresse ensuite mes remerciements respectueux aux membres du jury, Messieurs Clément Fortin et Adnène Hajji qui me font l'honneur d'examiner ce travail et d'en être les rapporteurs.

Je remercie également Monsieur Stephen A. Jarislowsky par l'intermédiaire de sa fondation, l'entreprise SNC-Lavalin et le CRSNG, pour l'aide financière dont j'ai pu bénéficier au sein de la Chaire de Recherche Jarislowsky / SNC-Lavalin en Gestion de Projets Internationaux. Je remercie aussi Mme Nathalie Marcoux et Monsieur Jean Paul Konecny pour leur implication au sein de la chaire de recherche.

Aussi, je tiens à remercier particulièrement Messieurs Joe Leone, Benoît Marcotte et Madame Dora Trifonova Stefanov, professionnels de SNC-Lavalin qui ont accepté de me rencontrer dans le cadre de mes recherches.

Je remercie les membres de l'équipe de Monsieur Pellerin, amis et collègues que j'ai eu l'opportunité de côtoyer, Alban, Andrée-Anne, Bassem, Houaida, Javad, Jean Baptiste et Xavier, qui par leur sympathie durant l'avancement de mon projet de recherche ont permis une collaboration des plus agréable.

Je tiens aussi à remercier ceux qui m'ont accompagné, soutenu, motivé, changé les idées, conseillé et qui ont ainsi joué un rôle de premier plan dans la bonne réalisation de cette maîtrise. Je pense tout particulièrement à Clara, Caroline, Eddy, Etienne, Guillaume, Jérôme, Léo, Loïc, Mathieu, Rémi, Romain, Stefanie et à tous mes amis qui sauront j'en suis sûr se reconnaître.

Enfin je ne peux finir sans remercier mes parents, Brigitte et Pierre, mon frère Jules, ainsi que l'ensemble de ma famille, sans qui rien n'aurait été possible et qui, malgré les difficultés ont su m'apporter le soutien nécessaire à la réalisation de mes différents projets.

RÉSUMÉ

L'évolution des processus d'affaire a encouragé les firmes internationales à se tourner vers le fonctionnement par projet pour maximiser leurs bénéfices. Ce fonctionnement s'est peu à peu généralisé. Cependant la gestion de projet demeure une discipline complexe qui nécessite des outils efficaces pour aider la prise de décision. Il faut aussi noter que la plupart des projets de grande ampleur sont aujourd'hui exécutés en régime accéléré. Les techniques d'accélération de projets sont multiples mais les professionnels ne disposent pas d'outils permettant d'évaluer clairement leur efficacité.

Reconnaissant ces nouvelles réalités et les besoins qui y sont associés, ce mémoire propose un processus d'accélération de projet à l'aide du chevauchement d'activités. Le processus propose l'identification des couples d'activités chevauchables, l'évaluation des paramètres de chevauchement, jusqu'à l'ordonnancement du projet. Une fois le processus établi, un modèle de base d'ordonnancement prenant en compte les différentes opportunités de chevauchement et permettant une résolution optimale a été présentée. Une méthode de génération d'instances de projet a parallèlement été développée pour permettre d'expérimenter ces méthodes de résolution. L'implémentation du modèle d'ordonnancement sur de nombreuses instances a permis d'identifier des paramètres capitaux pour l'efficacité du chevauchement comme technique d'accélération de projet.

Cependant, de nombreuses limites sont apparues suite à l'application de ce modèle, notamment le fait que le coût associé au chevauchement d'activités n'est pas considéré. Deux modèles alternatifs ont été envisagés, permettant d'améliorer les solutions obtenues. Le premier minimise le temps d'exécution tout en trouvant la solution à coût minimal alors que l'autre recherche le meilleur compromis entre le temps d'exécution et l'augmentation des coûts associés au chevauchement d'activités. Cependant ces modèles de programmation linéaire ne sont pas applicables à des projets de grande ampleur. Une méthode heuristique a été développée pour pallier ces limites, basée sur l'utilisation d'un progiciel de planification de projet répandu dans la pratique, qui permet de limiter les temps de calculs tout en assurant une bonne qualité aux solutions obtenues.

Les résultats obtenus sont très prometteurs et témoignent de l'intérêt à porter aux études sur le chevauchement. Les méthodes heuristiques ont permis d'obtenir des résultats de bonne qualité en un temps raisonnable.

Cependant, notre processus connaît certaines limitations qui ouvrent des perspectives de recherches pertinentes qui permettront d'augmenter l'applicabilité de notre modèle à des projets réels. Les principaux axes de recherches sont une meilleure considération des flux d'information, la prise en compte de l'incertitude, l'intégration à de nouveaux progiciels de gestion de projets ainsi que le développement d'une approche plus globale, permettant une application du processus à des projets de grande ampleur.

ABSTRACT

With globalization of markets, new business models have emerged and international companies started using project management techniques. However the management of projects remains complex and requires appropriate tools to support decisions. Further, most of the projects are executed in fast track. Project acceleration methods are numerous but project planners do not have effective tools to evaluate their decision.

This study proposes a process of accelerating project using the overlapping of activities. This process describes the identification of overlappable couples of activities, the assessment of overlap parameters and then the scheduling of the project. Once this process establish, we propose a basic scheduling linear programming model with overlapping opportunities. We also presented a generator of project instances with overlapping data. The implementation of the basic model in numerous instances allowed us to highlight the importance of three factors in the effectiveness of an overlapping strategy.

However this basic model did not consider cost augmentation due to an overlapping strategy. We propose two alternative linear programming scheduling models with different objectives which take into account cost influences. The first minimize the project completion time while considering cost increasing. The second search the best trade off between time reduction and augmentation of cost due to an overlapping strategy. These linear models are limited to small project because of the important computational time. We present a heuristic algorithm which finds acceptable solutions in reasonable time.

Results obtained are very promising and demonstrate the interest of considering overlapping as a complex project accelerating method. The proposed heuristic method is quite accurate but remains simple.

The study offers some interesting prospects for the future. Some of them would be the consideration of complex information flow and the variable nature of activity execution times, the improvement of the heuristic algorithm, and the development of a global approach for real project application.

TABLE DES MATIÈRES

REMERCIEMENTS	III
RÉSUMÉ.....	IV
ABSTRACT	VI
TABLE DES MATIÈRES	VII
LISTE DES TABLEAUX.....	X
LISTE DES FIGURES.....	XI
LISTE DES SIGLES ET ABRÉVIATIONS	XIII
LISTE DES ANNEXES.....	XV
INTRODUCTION.....	1
CHAPITRE 1 REVUE DE LITTERATURE	4
1.1 Introduction	4
1.2 Processus de planification de projet	4
1.3 Ordonnancement classique de projet.....	6
1.3.1 La décomposition en lots de travail.....	6
1.3.2 Ordonnancement de projet avec contraintes de ressources	9
1.4 Le chevauchement d'activités comme technique d'accélération de projet	12
1.4.1 Couplage de deux activités.....	13
1.4.2 Application aux projets industriels.....	16
1.5 Limitations	18
1.6 Objectifs de recherche	19
1.7 Conclusion.....	20
CHAPITRE 2 PROCESSUS D'ACCÉLÉRATION DE PROJET	21
2.1 Introduction	21

2.2	Méthodologie générale.....	21
2.3	Paramètres nécessaires à la définition des modes de chevauchement.....	26
2.4	Conclusion.....	27
CHAPITRE 3 MODELE OPTIMAL D'ORDONNANCEMENT DE PROJET AVEC CHEVAUCHEMENT D'ACTIVITÉS		28
3.1	Introduction	28
3.2	Hypothèses et notations.....	28
3.3	Le générateur de projet.....	30
3.3.1	Génération du réseau de projet avec contraintes de ressources	30
3.3.2	Paramètres de chevauchement	31
3.4	Modèle de base d'ordonnancement.....	34
3.5	Influence des caractéristiques du projet sur l'efficacité du chevauchement	36
3.5.1	Données expérimentales.....	36
3.5.2	Analyse des résultats	38
3.6	Limitations	45
3.6.1	Considérations des coûts associés au chevauchement d'activités.....	45
3.6.2	Temps de calcul.....	46
3.7	Conclusion.....	46
CHAPITRE 4 EXTENSIONS DU MODELE DE BASE		47
4.1	Introduction	47
4.2	Cas d'études considérés	47
4.3	Différentes approches d'optimisation	48
4.3.1	Prise en compte du coût associé au chevauchement	48
4.3.2	Compromis entre accélération et augmentation du coût de projet	50
4.3.3	Limitations	52

4.4	Approche de résolution approchée	53
4.4.1	Méthode.....	53
4.4.2	Analyse des résultats	57
4.4.3	Limitations	62
4.5	Conclusion.....	62
CHAPITRE 5 DISCUSSION		64
5.1	Introduction	64
5.2	Pertinence de l'étude	64
5.3	Limitations	66
5.3.1	Hypothèses	66
5.3.2	Méthodes d'ordonnancement	68
5.4	Améliorations envisageables	68
5.5	Conclusion.....	69
CONCLUSION		70
BIBLIOGRAPHIE		73
ANNEXES		77

LISTE DES TABLEAUX

Tableau 3.1 : Notations utilisées	28
Tableau 3.2 : Valeurs fixes dans le plan d'expérience	37
Tableau 3.3 : Différentes valeurs des paramètres du plan d'expérience	37
Tableau 4.1 : Diminution des coûts liés à l'optimisation des coûts	49
Tableau 4.2 : Comparaison entre les bénéfices associés au modèle de base et ceux issus du compromis entre accélération et augmentation des coûts reliés au projet	52
Tableau 4.3 : Temps de calculs (en seconde) associés à la résolution des modèles présentés	53
Tableau 4.4 : Nombre d'itérations nécessaires à l'obtention de la solution.....	60

LISTE DES FIGURES

Figure 1.1 : activités dépendantes	7
Figure 1.2 : activités indépendantes	8
Figure 1.3 : activités interdépendantes	8
Figure 2.1 : Exemple de différents modes de chevauchement possibles pour un couple d'activités (i,j)	22
Figure 2.2 : Processus d'accélération de projet avec modes de chevauchement	23
Figure 2.3 : Mécanisme de chevauchement de deux activités i et j dans le mode m	25
Figure 3.1 : Génération des données de chevauchement	33
Figure 3.2 : Modes de chevauchement pour $C_{max}=0,75$	38
Figure 3.3 : Diagramme de Pareto	40
Figure 3.4 : Impacts de RS sur le temps gagné	41
Figure 3.5 : Impacts de RS sur les coûts supplémentaires	41
Figure 3.6 : Impacts de R sur le temps gagné	42
Figure 3.7 : Impacts de R sur les coûts supplémentaires	43
Figure 3.8 : Impacts de C_{max} sur le temps gagné.....	44
Figure 3.9 : Impacts de C_{max} sur les coûts supplémentaires.....	44
Figure 4.1 : Primes et pénalités associées à l'exécution du projet	50
Figure 4.2 : Algorithme d'accélération de projet avec le chevauchement d'activités	54
Figure 4.3 : Gain et coût associés à l'exécution de l'algorithme avec la fonction d'évaluation du scénario 1.....	58
Figure 4.4 : Marges d'erreur associées à l'heuristique avec la fonction d'évaluation du scénario 1	59
Figure 4.5 : Gain et coût associés à l'exécution de l'algorithme avec la fonction d'évaluation du scénario 2.....	61

Figure 4.6 : Marges d'erreur associées à l'heuristique avec la fonction d'évaluation du scénario 2	
.....	61

LISTE DES SIGLES ET ABRÉVIATIONS

A	Ensemble des couples d'activités chevauchables
A(j)	Ensemble des prédécesseurs de j qui sont chevauchables avec l'activité j
B	Pourcentage du temps retouches associé à un temps de chevauchement
C	Complexity factor
C _{max}	Pourcentage maximum de chevauchement acceptable
CP	Ensemble des couples d'activités dépendant
d _j	Durée d'exécution de l'activité j
DSM	Design Structure Matrix
EF _j	Date de fin au plus tôt de l'activité j
LF _j	Date de fin au plus tard de l'activité j
m _j	Nombre de modes d'exécution possibles pour l'activité j
n	Nombre d'activités réelles
NP-difficile	Se dit d'un problème Non-déterministe Polynomial
P	Ensemble des couples d'activités non chevauchables
P(j)	Ensemble des prédécesseurs de j qui ne sont pas chevauchables avec l'activité j
Pred(j)	Ensemble des prédécesseurs de j
R	Proportion des couples d'activité chevauchables parmi CP
RCPSP	Resource Constraint Project Scheduling Problem
RF	Resource Strength
R _{jk}	Nombre d'unités de ressource k nécessaires par période pour l'exécution normale de l'activité j
r _{jm}	Durée des retouches associées au mode d'exécution m pour l'activité j
R _k	Quantité disponible constante pour la ressource k

RS	Ressource Factor
S	Ensemble des activités
SDP	Structure de découpage de projet
T	Durée maximale d'exécution du projet
$t = 0, \dots, T$	Périodes
WBS	Work Breakdown Structure
α_{ijm}	Pourcentage de la durée de chevauchement entre les activités i et j dans le mode d'exécution m, exprimé comme une fraction de d_j
θ_k	Coût horaire de la ressource k

LISTE DES ANNEXES

ANNEXE 1 : Article soumis à «International Journal of Project Organisation and Management».....	77
ANNEXE 2 : Article soumis pour la conférence du CIGI	91
ANNEXE 3 : Programme “Accélération de projet avec mode de chevauchement”.....	101

INTRODUCTION

La concurrence internationale et la globalisation dans les secteurs industriels sont des facteurs très contraignants pour les entreprises de nos jours. Afin de limiter leurs impacts, et demeurer ainsi compétitives dans le contexte actuel, les entreprises doivent adopter des méthodes de gestion adaptées.

A cet égard, le fonctionnement par projet figure parmi les méthodes de gestion les plus répandues. Cela consiste en l'allocation de ressources financières, humaines et matérielles, de manière à atteindre des objectifs précis dans le cadre d'exigences spécifiées à l'interne ou par un client. Un fonctionnement par projet nécessite l'établissement d'un échéancier, puis le suivi du déroulement des activités afin de respecter les exigences intrinsèques au projet. Cependant, une telle approche peut apparaître complexe. Cette complexité s'accroît d'autant plus lorsque l'on considère des projets internationaux en raison de l'apparition de nouvelles contraintes. Ces contraintes sont principalement liées à la taille importante des projets, à l'éloignement géographique des ressources, à la délocalisation des activités et à l'importance des activités de logistique.

Pour chaque projet, un échéancier précis doit être développé afin de faciliter les activités de suivi et de contrôle de projet. La réalisation de l'échéancier nécessite de nombreuses heures de travail afin de définir les activités à réaliser, les échanges d'information et les durées estimées des activités, pour enfin arriver au planning en tant que tel. L'échéancier est réalisé à partir de l'ordonnancement des activités sous différentes contraintes. Ces contraintes peuvent être des contraintes de main d'œuvre, de logistique, des contraintes liées aux conventions collectives, etc. Plusieurs méthodes d'ordonnancement ont été développées dans la littérature. Toutefois, leur efficacité n'a pas été démontrée dans le cadre de projet de grande ampleur. La réalité, dans laquelle une grande partie des projets dépassent les délais de complétion ou n'obtiennent pas les performances estimées, m'amène à penser que les outils utilisés ne sont pas adaptés.

De plus, dans la réalité, certains projets doivent être réalisés en régime accéléré. Ce régime est nécessaire pour maximiser les profits liés à la réalisation du projet en avançant la date de remise du projet. L'accélération de projet se veut aussi utile pour répondre aux écarts apparaissant lors de l'exécution d'un projet. Différentes techniques peuvent ainsi être adoptées pour permettre l'accélération d'un projet. Dans le cas des projets d'ingénierie, le chevauchement d'activités

constitue une approche très répandue dans la pratique. Cela consiste à commencer une activité à l'aide d'informations préliminaires, ce qui autorise de la commencer en avance par rapport au planning initial. Cette technique permet normalement l'accélération du projet mais implique la considération de nouvelles contraintes. En revanche, le chevauchement peut avoir de nombreux impacts sur le déroulement du projet et ainsi influencer la performance escomptée. Dans la pratique, les décisions sur les techniques d'accélération de projet se font souvent sur la base de recommandations *ad hoc*, liées par exemple à l'expérience des équipes de planification ou du client. Dans certains cas, les décisions prises ne sont pas toujours adaptées et peuvent entraîner l'effet inverse, c'est-à-dire retarder la date de clôture ou augmenter les coûts associés au projet. Il est donc nécessaire d'apporter une réponse efficace à la demande des entreprises en outils fiables de planification et de contrôle de projet.

Les nombreuses recherches dans le domaine de la planification de projet ignorent en grande partie les contraintes liées au contexte de chevauchement. D'une part, les approches de résolution exacte ne permettent pas la résolution de projet de grande ampleur, comprenant des milliers d'activités, comme c'est le cas pour la plupart des projets actuels. De plus, la plupart des études qui ont été faites sur les techniques d'accélération de projet ne considèrent pas les problématiques réelles des projets. Ainsi, elles ne considèrent que rarement les contraintes liées à la limitation des ressources disponibles, alors que leur gestion a un impact prépondérant sur le déroulement du projet.

Ainsi, malgré les progrès récents, les méthodes d'ordonnancement proposées pour la planification de projet de grande ampleur demeurent aujourd'hui incapables de prendre en compte les différentes stratégies d'accélération de projet, telles que le chevauchement d'activités, de manière efficace. Notre objectif de recherche est ainsi de proposer un processus d'ordonnancement de projet performant, qui intègre la nécessité de réaliser le projet en régime accéléré et qui soit capable de proposer un échéancier valide respectant les différentes contraintes liées au projet telles que les contraintes de ressources. Pour y arriver, notre démarche sera la suivante.

Dans un premier temps, une revue des approches proposées dans la littérature scientifique traitant des processus de planification de projet et d'exécution de projet en régime accéléré fut exécutée. Par la suite, un processus de planification de projet prenant en compte les opportunités de chevauchement fut proposé. Une première modélisation des alternatives de chevauchement fut

ensuite exposé. Enfin un modèle optimal d'ordonnancement de projet déterministe prenant en compte les opportunités de chevauchement d'activités fut développé. Le modèle développé fut testé sur de nombreux cas d'études dans le but de valider la démarche proposée. Ensuite, le modèle de base fut revu de façon à intégrer les problématiques de coûts et de pouvoir traiter des projets de plus grande taille et ainsi, s'approcher des cas rencontrés dans le milieu industriel. L'analyse des résultats sur ces différentes modélisations a permis de valider l'approche. Finalement, ce mémoire termine par une discussion et une analyse de la pertinence de cette étude vis-à-vis les problématiques industrielles, en plus de mettre en avant différentes opportunités futures de recherche.

CHAPITRE 1 REVUE DE LITTÉRATURE

1.1 Introduction

Au cours des dernières années, l'intérêt pour la gestion de projet n'a fait qu'augmenter. Il a permis le développement de nombreuses approches de planification et de suivi de projet. Dans le même temps, les exigences des clients et de la concurrence ont mené les projets à s'exécuter de plus souvent en régime accéléré, ou *fast-tracking* en anglais. A travers la revue de littérature suivante, je m'attacherais à positionner notre approche parmi les différents travaux de recherches. Pour cela, les différents processus de planification de projet et les différentes techniques d'ordonnancement de projet qui ont été développées seront abordés. Ensuite, les études menées sur le chevauchement d'activités, une technique d'accélération de projet, sera présentée. Finalement, ce chapitre se termine par une analyse critique des lacunes des recherches antérieures dans le but de positionner notre contribution scientifique.

1.2 Processus de planification de projet

La gestion d'un projet de grande envergure nécessite l'exécution de différents processus. Le premier d'entre eux est le processus de démarrage du projet qui comprend les études de pré-faisabilité et de faisabilité, puis le processus de planification, à partir duquel il sera possible d'établir le plan de gestion du projet. Il en suit le processus d'exécution du projet qui se termine au début de l'exploitation. Le processus de surveillance et de contrôle de projet surveille la planification et l'exécution du projet en mesurant la performance en temps réel mais aussi en prévenant les risques (Project Management Institute, 2008). Nos recherches visent plus spécifiquement les processus de planification.

Le processus de planification de projet comprend l'exécution de plusieurs tâches (Hegazy, 2002) : tout d'abord l'élaboration de la Structure de Découpage de Projet (SDP) ou *Work Breakdown Structure* (WBS). L'élaboration de la SDP consiste à diviser le projet en éléments plus petits, comme des lots de travail ou des activités afin de décrire l'ensemble du projet (Kerzner, 2003). La SDP est un élément essentiel pour la bonne gestion du projet, car elle permet d'allouer des heures de travail et un budget spécifique à chaque tâche de la SDP. Elle est donc importante vis-à-vis de la gestion du temps d'exécution et du budget. Une mauvaise SDP rendra

la gestion du projet et la mesure de la performance beaucoup plus difficiles du fait du manque de données. Une fois la SDP mise en place, il est possible de définir les dépendances entre les tâches. Cela se fait à l'aide de rencontres avec les responsables de chaque tâche en analysant les données d'entrée nécessaires à l'exécution des activités. Ce processus est primordial pour permettre une bonne planification par la suite.

La connaissance des dépendances permet d'établir le réseau de projet. Celui-ci relie les tâches dépendantes entre elles et permet de créer une suite logique de tâche qui reflète l'exécution correcte du projet. C'est ce réseau qui définira le déroulement du projet et l'enchaînement des tâches. Cependant, il n'a aucune signification temporelle.

Afin de prendre en compte le caractère temporel de l'exécution du projet, une estimation des durées des activités et des ressources nécessaires à une exécution normale de ces activités doit être faite en amont. Cette phase d'estimation est cruciale pour la réussite du projet par la suite. Ensuite, il est possible de réaliser l'échéancier en tant que tel en allouant des dates de début et de fin pour chaque activité.

Une fois les données acquises, les méthodes d'ordonnancement de projet avec contraintes de ressources sont multiples. Elles permettent une résolution exacte ou approchée, en un temps plus ou moins important en fonction des caractéristiques du projet et de la précision souhaitée. Une fois le planning initial développé, certains projets nécessitent toutefois d'être exécutés en régime accéléré afin de respecter les exigences du client.

Pour sa part, le chevauchement d'activités se veut une technique commune pour réduire la durée d'exécution du projet. Cela consiste à commencer une activité avant d'avoir toutes les informations nécessaires à la réalisation de celle-ci. Cette technique, très répandue dans un environnement d'ingénierie simultanée, permet d'accélérer l'exécution du projet. Cette stratégie peut être adoptée dans le cadre du processus normal de planification de projet ou lors de son exécution.

Sur le plan scientifique, il ressort deux grands axes de recherches : les approches d'ordonnancement de projet avec contraintes de ressources et les techniques de réduction du temps d'exécution. Bien que les techniques d'ordonnancement de projet soient fort nombreuses, peu s'appliquent à des projets avec chevauchement et leur portée se limite bien souvent aux projets de petite taille.

1.3 Ordonnancement classique de projet

L'ordonnancement de projet classique nécessite d'abord la décomposition du projet en activités. Puis, dans un second temps, les activités sont programmées en fixant des dates de début et de fin afin de générer un échéancier réaliste guidant l'exécution du projet. Ces deux éléments sont traités ci-après.

1.3.1 La décomposition en lots de travail

Une fois l'envergure du projet définie, le premier travail pour les planificateurs est d'identifier les activités en lots de travail (Project Management Institute, 2008). Pour cela, ils ont besoin de tenir compte des facteurs environnementaux de l'entreprise, de l'actif organisationnel et de l'énoncé du contenu du projet.

La décomposition d'un projet se fait de deux manières possibles : formelle ou intuitive (Bhatt, 1988). Les techniques de décomposition formelles se basent sur une représentation mathématique et systématique du problème. Les techniques intuitives se basent plutôt sur l'expérience de l'entreprise et de l'équipe de planificateurs et sur leur habilité à bien appréhender le projet.

1.3.1.1 Techniques de base de décomposition

Les techniques de décomposition sont utiles lors d'une décomposition intuitive du projet et permettent d'aborder de manière formelle le problème. Certains papiers portent un regard particulier sur les techniques de décomposition. La taille des lots de travail, c'est-à-dire le temps de travail moyen nécessaire pour compléter un lot, est un paramètre pris en compte. Pour un type d'activités donné, il est possible de trouver une taille optimale de lot de travail. Le niveau de détail optimal de la SDP dépend de la nature et de la taille du projet (Jung et Woo, 2004) : une décomposition fine permet de contrôler plus en détail les dépenses liées au projet, mais nécessite un travail de gestion beaucoup plus important tout au long de l'exécution du projet. Pour les projets de grande ampleur, on privilégie une décomposition grossière en ce qui concerne le budget, sauf dans les cas où l'on veut privilégier de petites équipes de projet (Hendrickson et Au, 1989). Un moyen de mieux contrôler les budgets tout en minimisant l'effort de gestion est d'utiliser une structure de décomposition du budget différente de la structure de décomposition du projet.

Une fois les lots de travail définis et leurs dépendances identifiées, il s'agit de les regrouper en groupes de travail cohérents. On s'occupe alors de diminuer au maximum la densité d'interactions entre les groupes de travail. La densité d'interactions est définie par le degré de dépendance entre les groupes de travail du projet. Kusiak (1993) propose une méthode pour regrouper les lots de travail en groupes indépendants. On représente les interdépendances à l'aide d'une matrice carrée. Les lignes et les colonnes de la matrice représentent les activités. Les valeurs binaires dans la matrice représentent la présence ou non d'échange d'information entre les activités. Dans un premier temps, les groupes de travail n'apparaissent pas clairement dans cette matrice. On applique alors l'algorithme d'identification de groupes (Kusiak et Chow, 1987) à la matrice pour obtenir une nouvelle matrice dans laquelle on peut identifier clairement les groupes de travail, lesquels ont une densité d'interactions minimale. A partir de ce modèle, les chercheurs ont développé les matrices de décomposition de projet, ou *Design Structure Matrix* (DSM). Je vais présenter leurs intérêts et leurs applications dans le cadre de l'ordonnancement de projet dans la section suivante.

1.3.1.2 Les matrices de décomposition

Les matrices de décomposition ont été introduites par Steward (1981). Il définit les matrices de dépendances de manière binaire, 0 lorsqu'il n'y a pas d'interaction entre les activités, 1 lorsque celles-ci sont liées. Ces matrices permettent de modéliser tous les types de dépendances entre les activités :

- Activités dépendantes (séquentielles) :

	activité i	activité j
activité i	-	0
activité j	1	-

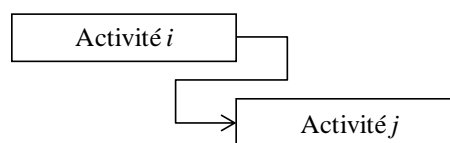


Figure 1.1 : activités dépendantes

Les activités dépendantes sont des activités reliées par une relation de précédence (fin à début, fin à fin, début à début,...). La relation est unidirectionnelle, l'activité j est ici dépendante de l'activité i .

- Activités indépendantes :

	activité <i>i</i>	activité <i>j</i>
activité <i>i</i>	-	0
activité <i>j</i>	0	-

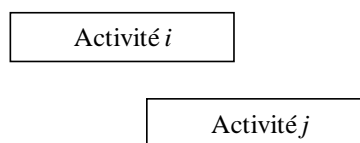


Figure 1.2 : activités indépendantes

Les activités indépendantes sont des activités qui peuvent être exécutées en parallèle sans qu'il n'y ait d'impact sur l'une ou l'autre.

- Activités interdépendantes (couplées) :

	activité <i>i</i>	activité <i>j</i>
activité <i>i</i>	-	1
activité <i>j</i>	1	-

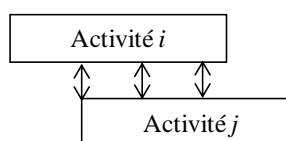


Figure 1.3 : activités interdépendantes

Les activités interdépendantes sont dépendantes l'une de l'autre. Elles doivent être réalisées en parallèle et se transmettent des informations tout au long de leur exécution.

On peut donc modéliser l'ensemble des dépendances entre les différentes activités du projet. Regrouper les activités dépendantes entre elles permet de limiter les problématiques de communication et de gestion et donc de gérer plus efficacement le projet.

Ce modèle permet d'identifier rapidement et de regrouper les groupes d'activités dépendantes, et donc d'optimiser la gestion des flux d'information. Les trois principaux avantages des matrices de décomposition sont (Chen et Li, 2003) :

- la possibilité de les appliquer à des modèles de complexité et de taille importante ;
- la clarté de la modélisation qui permet une communication claire et accessible à tous ;
- la possibilité de traiter les données facilement à l'aide d'algorithmes.

Ce modèle a été repris et amélioré dans de nombreux papiers. Eppinger et al. (1994) proposent un modèle plus réaliste avec des niveaux de dépendances variables, ainsi qu'un algorithme permettant de réorganiser efficacement les activités. Cela permet de créer des groupes de travail cohérent de manière automatisée.

La nature des DSM permet de choisir les critères de regroupement. Cette possibilité a été étudiée par Browning (2001). Dans ce papier est évoquée la possibilité de regrouper les activités par différents critères (échange d'information, matériel utilisé, énergie nécessaire...) à l'aide de DSM. Les regroupements permettent d'adapter la décomposition au contexte et aux exigences du projet, et donc d'optimiser celle-ci.

Dans leur papier, Chen et Li (2003) décrivent une méthode d'optimisation des DSM. Ils proposent de ne plus utiliser de matrice binaire. La méthode utilisée permet de développer des matrices de couplage numérique afin de regrouper les activités plus efficacement. Dans leur papier, ils montrent à travers un exemple sur des matrices de petite taille (20x20) que cette méthode permet d'améliorer significativement l'efficacité du regroupement et de gérer plus efficacement le projet par la suite.

Un algorithme d'optimisation du chemin critique a aussi été développé à l'aide des matrices de décomposition (Fayez et al., 2003). Il propose de décomposer chaque activité en deux sous-activités, « *Information flow* » et « *Work Flow* ». Ensuite, on optimise la matrice de décomposition de manière à limiter au maximum les retours d'informations. Les activités dont dépendent beaucoup d'échanges d'informations sont divisées en sous-activités pour limiter leurs impacts.

L'utilisation des matrices de décomposition dans un processus global de planification permet de regrouper efficacement les activités et d'obtenir un réseau d'activités dont les flux d'information sont *a priori* unidirectionnels. Cette caractéristique permet de faciliter par la suite l'ordonnancement et le suivi de projet lors de l'exécution.

1.3.2 Ordonnancement de projet avec contraintes de ressources

Une fois le réseau de projet et la séquence d'activités déterminée, il est nécessaire d'établir le planning en tant que tel. Cette étape passe par l'ordonnancement du projet en considérant l'ensemble des contraintes intrinsèques au projet. Le problème d'ordonnancement de projet consiste à ordonnancer un ensemble d'activités liées par des relations de précédences dans le but de minimiser le temps de réalisation du projet sous des contraintes de précédence et d'utilisation de ressources. Les problèmes d'ordonnancement avec contraintes de ressources, appelés communément RCPSP (*Resource-Constrained Project Scheduling Problem*), ont été

abondamment étudiés dans les dernières années. Ces problèmes ont été introduits par Pritsker (1969). Depuis, de nombreuses recherches visent à optimiser la planification des activités pour minimiser la durée totale du projet. Les problèmes de type RCPSP sont NP-difficiles au sens fort (Blazewicz et al., 1983). Durant les dernières années, le RCPSP est devenu un problème classique de planification et de nombreux états de l'art ont été réalisés (Brucker et al., 1999; Hartmann et Briskorn, 2010). On peut y différencier deux types d'approches, présentées par la suite : la programmation linéaire et les approches de résolution approchée.

1.3.2.1 Programmation linéaire

Les méthodes exactes ont été les premières approches de résolution développées. La plupart se base sur la résolution de problème de programmation linéaire en nombres entiers (Pritsker et al., 1969). Cependant, une telle approche ne permet pas de traiter facilement des projets d'une taille supérieure à 60 activités car la linéarisation des contraintes de ressources complexifie énormément le problème d'ordonnancement de base.

Des modèles d'ordonnancement multi-mode ont aussi été développés pour envisager plusieurs scénarios d'exécution pour les activités. Chaque mode représente alors une alternative d'exécution pour l'activité. Une activité, exécutée dans le mode m , doit être terminée dans le mode m et on associe une durée d'exécution, des relations de précédence et une consommation spécifique des ressources à chaque mode m . Le problème d'ordonnancement multi-mode est aussi NP difficile (Brucker et al., 1999).

La plupart des modèles de programmation linéaire en nombres entiers propose une résolution par séparation et évaluation progressive, ou *branch-and-bound* (Demeulemeester et Herroelen, 1997). Cela consiste à énumérer et parcourir un ensemble de solutions selon une arborescence, définie en fonction des contraintes du projet. L'espace des solutions est alors réduit et séparé en sous-problèmes, chacun ayant leur ensemble de solutions réalisables.

Récemment, des approches hybrides ont été développées afin de permettre l'adaptation à des projets de plus grande envergure. Elles combinent les méthodes de séparation et d'évaluation progressives et des méthodes heuristiques (Liu et al., 2005). Mais de manière générale, les méthodes de résolution exacte sont assez peu utilisées en pratique dans l'ordonnancement de projet à cause de la limitation due à la taille et à la complexité des projets réels (Herroelen, 2005).

Cependant, ces approches permettent la résolution exacte des problèmes d'ordonnancement de petite taille et offrent l'opportunité de tester la qualité d'approches de résolution approchée.

1.3.2.2 Approches de résolution approchée

La taille des projets considérés est un facteur limitant l'efficacité des approches de résolution. Les recherches se sont donc tournées naturellement vers des méthodes de résolution approchée, dites heuristiques ou métaheuristiques. Ces dernières sont développées de manière à proposer un échéancier réalisable, respectant l'ensemble des contraintes de projet, dans un temps de calcul raisonnable. Les solutions obtenues ne sont certes pas optimales, mais compte tenu des incertitudes liées à l'estimation des données de projet, elles sont suffisantes (Hartmann 1999).

La plupart des méthodes heuristiques utilisent des règles de priorités. Les deux principaux schémas de génération d'échéancier sont les schémas sériel et parallèle, qui à chaque itération sélectionnent les activités qui peuvent être planifiées. L'approche sérielle considère l'activité comme incrément alors que l'approche parallèle utilise une approche temporelle où l'incrément correspond à une période temporelle (jour, semaine, ...). A chaque itération, les activités éligibles sont ordonnées en fonction de différentes règles de priorité afin d'établir un échéancier valide. Le choix des règles de priorité a une influence déterminante sur la qualité de la solution finale obtenue (Hartmann, 1999 ; Kolisch et Hartmann, 2006). Les progiciels de gestion de projet utilisent dans l'ensemble des heuristiques avec des règles de priorité simples qui ne donnent pas forcément des échéanciers de bonne qualité. La qualité des solutions est soumise à une grande variabilité (Herroelen, 2005).

Kolisch et Hartmann (2006) ont montré que l'approche de génération parallèle est plus adaptée pour les projets de grande taille comportant de nombreuses activités. Il faut cependant noter qu'aucune des deux méthodes ne se dégage vraiment quant à la qualité des résultats. Des études ont aussi montré que la considération de règles de priorité dynamique, évoluant en fonction des allocations précédentes, permet d'obtenir des résultats intéressants (Pellerin, 1997).

Enfin, des approches métaheuristiques ont aussi été développées de manière à améliorer la qualité des solutions et à éviter les minimums locaux. On peut citer par exemple les méthodes d'algorithmes génétiques et de recherche tabou, qui ont suscité beaucoup d'intérêt (Kolisch et Hartmann, 2006). Ces approches permettent d'obtenir des solutions toujours réalisables en un

temps acceptable, tout en assurant une meilleure qualité de solution que les méthodes heuristiques simples.

Ces méthodes de planification sont applicables dans le cadre d'un contexte d'exécution classique de projet. Cependant, dans la réalité des projets internationaux, les enjeux financiers amènent souvent à exécuter les projets en régime accéléré. Cela peut permettre à la fois de limiter les risques de retards ou de respecter les exigences des clients. Parmi les différentes techniques d'accélération de projet, on compte le chevauchement d'activités, la compression d'activité, et la substitution d'activité. La compression d'activité consiste à allouer des ressources supplémentaires à une activité pour accélérer son déroulement. La substitution d'activité consiste, quant à elle, à remplacer une activité par une autre, moins longue ou moins coûteuse. Cela peut correspondre, par exemple, à un changement de procédé dans l'exécution du projet. Le chevauchement d'activités est aussi une technique classique d'accélération de projet. Dans la section suivante, je vais présenter le chevauchement d'activités et exposer les différentes études le concernant.

1.4 Le chevauchement d'activités comme technique d'accélération de projet

Le chevauchement est aujourd'hui une technique couramment utilisée lors de l'exécution d'un projet en régime accéléré. L'approche du chevauchement d'activités est complexe à cause des nombreux paramètres qui interagissent entre les activités. Le chevauchement entraîne souvent des remaniements et modifications, provoquant une augmentation des coûts reliés. Il a été confirmé par une étude statistique (Terwiesch et Loch, 1999) que cette technique permet de réduire la durée d'exécution du projet. Les bénéfices sont d'autant plus importants si on arrive à réduire les incertitudes rapidement. Cette technique peut notamment être utilisée pour limiter les risques de retards lors de l'exécution du projet. Cependant, il est avéré que le chevauchement entraîne des remaniements, modifications et une augmentation des coûts d'exécution directement liés au projet (Roemer et al., 2000). Il existe un niveau optimal de chevauchement entre deux activités, permettant un compromis entre une diminution du temps et une augmentation du coût d'exécution (Terwiesch et Loch, 1999). Lin et al. (2009) ont remarqué que les entreprises déterminent la plupart du temps les stratégies de chevauchement sur une base *ad hoc* et sans considérer ni les interactions entre les activités, ni les retouches dues au chevauchement et à l'exécution de l'activité avec des informations préliminaires, ni les contraintes de ressources qui

peuvent être importantes au sein d'un projet. D'où l'importance de développer un modèle adaptable en milieu industriel.

Les modèles proposés dans la littérature diffèrent par leurs approches spécifiques à un couple d'activités ou à des considérations plus globales, adaptées aux projets industriels.

1.4.1 Couplage de deux activités

De nombreux modèles se contentent dans un premier temps de ne considérer qu'un couple d'activités : une activité en amont dont dépend la réalisation de l'activité en aval. Cela permet une approche précise du problème mais nécessite une connaissance avancée des activités et de leurs interactions.

Un cadre pour les stratégies de chevauchement (Krishnan et al., 1997) prend comme paramètres l'évolution de l'activité en amont et la sensibilité au changement de l'activité en aval. Dans ce modèle, l'évolution de l'activité reflète le niveau d'information pouvant être transmis à l'activité en aval. L'influence de la vitesse d'évolution de l'activité en amont sur le gain effectif du chevauchement a été démontrée de manière statistique (Terwiesch et Loch, 1999). La sensibilité est le temps de travail nécessaire pour pallier un changement de l'activité précédente. Ces deux paramètres varient au cours de l'avancement des activités. Les courbes d'évolution de ces deux indicateurs permettent de définir des stratégies de chevauchement adaptées aux couples d'activités. Ces stratégies considèrent les échanges d'informations, préliminaires ou finales, et les niveaux de chevauchement. Krishnan définit ainsi quatre grandes stratégies de chevauchement : le chevauchement itératif, le chevauchement distributif, le chevauchement fractionnel et le chevauchement préemptif ; chacune s'adaptant à un couple sensibilité/évolution donné.

Lorsque la sensibilité au changement est faible et que la vitesse d'évolution est lente, on peut appliquer le chevauchement itératif : on peut commencer l'activité en aval à l'aide d'information préliminaire et intégrer les informations au cours de l'évolution. Les échanges d'information ne se finissent que lorsque l'activité en amont se termine.

Le second cas s'applique lorsque la sensibilité est faible et la vitesse d'évolution rapide, on applique alors le chevauchement distributif : on commence l'activité aval avec les informations préliminaire, puis rapidement on intègre des informations finalisées.

Le troisième cas s'applique aux sensibilités élevées et à une évolution lente. Il est alors difficile de chevaucher les activités, on parle de chevauchement fractionnel, qui revient finalement à l'absence de chevauchement : on crée des sous-activités permettant un échange d'information finalisée afin d'éviter les changements.

Enfin, le quatrième cas s'applique aux sensibilités élevées et aux évolutions rapides, on applique alors le chevauchement préemptif : on fait un seul échange d'information finale avant la fin de l'activité en amont, permettant de commencer l'activité aval plus tôt, tout en s'assurant qu'il n'y aura pas de changement par la suite.

Ce modèle a été repris et amélioré en s'attachant notamment à identifier les différentes stratégies de chevauchement et les facteurs influençant l'évolution et la sensibilité des activités (Bogus et al., 2005). Dans leurs travaux, ces auteurs ont identifié huit stratégies permettant d'améliorer les approches de chevauchement :

- Bloquer rapidement des éléments de *design* afin de pouvoir partager rapidement des informations définitives avec les activités en aval. Le design décrit les caractéristiques techniques des éléments d'ingénierie.
- Sur-dimensionner l'activité en aval afin de pouvoir pallier un changement dans le *design* de l'activité en amont.
- Compléter rapidement les informations préliminaires pour permettre à l'activité en aval de commencer avant la fin de l'activité en amont.
- Prototyper afin d'obtenir rapidement un modèle, et permettre de commencer avant la fin complète du *design*.
- Limiter l'optimisation et les itérations pour passer plus rapidement les informations aux activités suivantes.
- Utiliser des outils/produits standardisés.
- Composer des ensembles de solutions en parallèle.
- Décomposer une activité en plusieurs activités évoluant plus rapidement, cela permettra de diminuer la sensibilité. Cette stratégie de dernier recours permet ensuite d'appliquer les stratégies précédentes aux nouvelles activités.

La courbe d'évolution d'une activité est caractérisée par le niveau d'optimisation et la prise en compte de plusieurs alternatives, le nombre de contraintes à satisfaire, le niveau d'échange d'information et la possibilité de standardisation. La sensibilité est tributaire de la sensibilité des contraintes, des *inputs* et de l'intégration. Ces quatre facteurs pour l'évolution, et trois pour la sensibilité ont permis de développer un cadre avancé de stratégie de chevauchement proposant des stratégies (parmi les huit présentées précédemment) adaptées aux types d'évolution et de sensibilité des activités concernées.

Dans la définition de l'évolution d'une activité, les activités de communication sont primordiales car l'avancement ne peut se mesurer qu'après une action de communication. Deux modèles se distinguent : les premiers considèrent que la communication est « gratuite » et instantanée. Dans ce cas précis, il n'est pas nécessaire de planifier et d'organiser les échanges d'information. Roemer et al. (2000) ont développé avec cette approximation le premier modèle permettant d'obtenir un compromis entre la diminution du temps d'exécution et une augmentation du budget donnée. Le modèle de Krishnan et al. (1997) présente les stratégies de chevauchement en ignorant les coûts et délais de la communication entre activités dépendantes.

D'autres, au contraire, considèrent que la communication implique un coût et un temps d'exécution. Le cas échéant, on pourrait considérer que les modifications sur l'activité en amont seraient immédiatement incorporées à l'activité en aval (Loch et Terwiesch, 1998), ce qui n'est pas réaliste. Dans le même papier de Loch et Terwiesch, les auteurs montrent que plus la communication est rapide après un changement, moins l'impact de celui-ci est important.

Ces modèles restent principalement quantitatifs et ils ne proposent pas de solution optimale de chevauchement. L'étude de Lin et al. (2009) modélise de façon plus générale et plus complexe le problème de couplage de deux activités avec une fonction objectif alliant temps d'exécution et de coût. L'approche proposée est analytique.

Cependant, l'ensemble des études présentées dans cette section ne propose pas une approche globale de gestion de projet en se réduisant seulement aux interactions entre deux activités. Aussi, l'application des modèles présentés nécessite des études poussées sur les interactions et échanges d'information entre les activités, ce qui limite leur application à des projets de petite taille.

1.4.2 Application aux projets industriels

Des modèles apportent une vision plus globale du problème en considérant non pas un couple d'activités, mais plutôt un projet dans sa globalité. De telles considérations nécessitent d'autres méthodes d'analyse que l'on peut regrouper en deux groupes. Tout d'abord, les premiers considèrent qu'il est possible d'estimer les quantités de modifications qu'il faudra faire à cause du chevauchement d'activités (Roemer et al., 2000) ; ces modèles sont donc des modèles d'optimisation déterministes. Le second groupe de modèles considère qu'il n'est pas possible en pratique d'estimer les quantités de travail supplémentaires ; on utilise donc la simulation pour parvenir à un résultat plus proche de la réalité. La simulation permet en effet de prendre en compte la variabilité de certains paramètres intrinsèques au projet. Cependant, cette technique ne permet pas de produire un échéancier diffusable au sein d'une équipe de projet.

Dans tous les cas, ces modèles consistent à modéliser l'évolution des activités, l'impact du chevauchement sur l'objectif (coût et/ou temps d'exécution) ainsi que toutes les variables du projet.

1.4.2.1 Modèles d'optimisation

Le modèle proposé par Gerk et Qassim (2008) vise à accélérer un projet à l'aide des trois techniques les plus répandues d'accélération de projet : le chevauchement d'activités, la compression et la substitution d'activité. Il propose de trouver un échéancier à coût minimal en combinant ces trois méthodes d'accélération de projet. Le problème est posé sous la forme d'un programme linéaire en nombres entiers. Les données d'entrée sont nombreuses et augmentent rapidement avec le nombre d'activités, ce qui limite l'applicabilité d'un tel modèle. De la même façon, Liberadore et al. (2004) ont développé un modèle d'optimisation en nombres entiers visant à réduire la durée totale du projet en intégrant le chevauchement d'activités.

Malgré leur optimalité, ces techniques d'optimisation nécessitent souvent des temps de calculs importants, même pour des projets de petite taille. Cela est dû au caractère NP-difficile du problème d'ordonnancement avec contraintes de ressources. Les applications sont donc limitées à cause de la difficulté de résolution des gros problèmes. Les approches de résolution approchées permettent d'aborder des projets de plus grande taille mais ne garantissent pas l'optimalité des solutions. Parmi ces approches, on différencie les modèles heuristiques déterministes, qui

proposent de chercher une solution approchée, des modèles de simulation stochastiques, qui utilisent la simulation afin de prendre en compte les incertitudes sur un ou plusieurs paramètres du projet (durée des activités, temps de retouches, ...). Les méthodes de simulation seront présentées dans la section suivante.

Concernant les méthodes d'optimisation approchées, Roemer et al. (2000) ont développé un algorithme débouchant sur un compromis entre le temps d'exécution d'un projet de développement de produit et le coût de celui-ci. Il permet d'obtenir une réduction du temps d'exécution pour une augmentation de budget donnée, ou l'inverse (l'augmentation du budget en fonction de la durée espérée du projet). Appliquée à un exemple de projet industriel, cette méthode permet d'obtenir un gain en temps de 34% pour une augmentation du budget de 10%.

1.4.2.2 Modèles de simulation

Certains papiers utilisent des modèles de simulation afin de mieux refléter le caractère aléatoire de la réalité. Cho et Eppinger (2005) ont développé un modèle très complet de simulation avec des durées d'activités stochastiques, modélisant les chevauchements d'activités, itérations et retouches. Ils ont aussi pris en compte les contraintes de ressources. Ils ont montré que les contraintes de ressources peuvent retarder le chevauchement de certaines activités et donc retarder la date de fin du projet.

Wang et Lin (2009) ont développé un modèle de chevauchement stochastique pour prendre en compte les risques lors de l'ordonnancement. Leur modèle de simulation considère les itérations et les probabilités de retouches. Les itérations sont définies dans la littérature comme les interactions entre plusieurs activités de *design* qui nécessitent des retouches dues à des retours d'information de la part des activités en aval. A chaque itération, il faut retoucher une partie du travail déjà fait dans l'activité. Les quantités de retouches faites après chaque itération et après le chevauchement d'activités sont déterminées de manière probabiliste afin de prendre en compte le caractère aléatoire du temps de retouche. Cependant, ce modèle ne prend pas en compte les contraintes de ressources. Il n'est pas adapté à des problématiques réelles de projet avec ressources limitées.

Tous ces travaux considèrent les alternatives de chevauchement de manière continue en permettant toutes les possibilités de chevauchement. Le temps de chevauchement admissible y est alors défini entre une borne inférieure et une borne supérieure.

Aussi, de tels modèles ne permettent pas d'obtenir un échéancier diffusable. Ils permettent toutefois d'obtenir l'estimation la plus probable du temps d'exécution du projet. Cette estimation ne peut être utilisée comme un échéancier pour les fournisseurs ou les sous-traitants car elle repose sur des paramètres stochastiques qui ne peuvent être diffusés au sein du projet. C'est pourquoi dans la suite de cette étude, on considère que les données utilisées sont déterministes.

1.5 Limitations

Au cours de cette revue de littérature, différentes problématiques relatives à notre sujet ont été abordées, à savoir l'ordonnancement classique de projet et l'utilisation du chevauchement d'activités comme technique d'accélération de projet. On a ainsi pu démontrer l'importance de la séquence d'activités et l'existence d'algorithmes permettant d'obtenir une séquence d'activités unidirectionnelle. Il fut aussi démontré l'existence de nombreuses méthodes d'ordonnancement, exactes ou non, qui prennent en compte les contraintes liées aux ressources humaines. Aussi, dans une optique d'accélération de projet, des modèles qui permettent de prendre en compte les paramètres de chevauchement définis au préalable ont été présentés. Toutefois, aucun modèle ne propose un processus général d'accélération de projet à travers le chevauchement, proposant une méthode d'acquisition des données de chevauchement et une méthode d'ordonnancement.

Il apparaît donc que les modèles d'accélération de projet avec chevauchement d'activités ne vont pas dans le sens de la convergence entre les hypothèses et la réalité des problématiques de projet. Du côté des hypothèses utilisées, de nombreux papiers ne prennent pas en considération les nombreuses contraintes liées aux ressources. Aussi, la plupart des modèles de chevauchement ne limite les possibilités de chevauchement que par une borne inférieure et une borne supérieure. Dans la réalité, toutes les configurations de chevauchement ne sont pas envisageables, on parle plutôt de modes de précedence ou de chevauchement possible qui définissent les configurations envisageables de chevauchement. La plupart des modèles étudiés ne sont applicables qu'à un nombre assez limité d'activités.

1.6 Objectifs de recherche

En regard de la revue de la littérature, les questions de recherche suivantes apparaissent :

- Comment peut-on définir et obtenir les paramètres nécessaires pour quantifier efficacement l'impact du chevauchement d'activités sur l'exécution d'un projet ?
- Comment peut-on intégrer le chevauchement d'activités à un modèle d'ordonnancement ?

Compte tenu de l'importance de ces deux questions de recherche, l'objectif général de ce mémoire consiste à développer un modèle de planification de projet en mode d'accélération en considérant les chevauchements d'activités. Plus précisément, les objectifs spécifiques suivants précisent le cadre de cette recherche :

- Applicabilité : le modèle doit proposer un processus simple et applicable dans un milieu industriel.
- Validation du modèle : le modèle proposé doit pouvoir être testé sur des cas théoriques variés modélisant l'ensemble des contraintes d'un projet.
- Performance du modèle : le modèle doit permettre d'obtenir des solutions de qualité acceptable en un temps de résolution raisonnable.

Les hypothèses scientifiques originales de notre contribution à la recherche (HSOC) sont les suivantes :

- HSOC n°1 : L'établissement d'un processus global d'accélération de projet à travers le chevauchement d'activités permet d'obtenir un planning plus réaliste, et ainsi un contrôle plus efficace du déroulement du projet.
- HSOC n°2 : La détermination des données de chevauchement permet de refléter fidèlement la réalité des interactions entre les activités.

L'originalité de ces hypothèses découle du fait qu'à notre connaissance, aucune approche globale ne considère le processus d'accélération de projet à travers le chevauchement d'activités à un niveau global. Aussi, aucune approche d'ordonnancement n'a modélisé le chevauchement de manière réaliste et applicable en milieu industriel.

Sur le plan expérimental, une méthodologie de génération de projet avec contraintes de ressources et paramètres de chevauchement sera proposée. J'explorerai sur les exemples générés différentes approches d'ordonnancement et je comparerai à la fois l'impact du chevauchement comme technique d'accélération de projet en fonction des paramètres de projet, ainsi que les performances de méthodes d'ordonnancement.

1.7 Conclusion

Après avoir passé en revue l'état de la recherche sur les modèles ayant trait à l'ordonnancement et à l'accélération de projet à l'aide du chevauchement d'activités, je vais présenter le modèle de planification général proposé dans le prochain chapitre. Ce modèle couvre les processus de création de la séquence d'activités, la détermination des paramètres de chevauchement et l'ordonnancement du projet.

CHAPITRE 2 PROCESSUS D'ACCÉLÉRATION DE PROJET

2.1 Introduction

Ce chapitre vise à présenter le processus général d'accélération de projet à l'aide du chevauchement d'activités. L'objectif est de proposer une méthodologie pour considérer le chevauchement et ses impacts sur la durée d'exécution du projet de manière analytique. On présentera dans un premier temps la méthodologie générale puis nous mettrons en avant les paramètres nécessaires à son exécution.

2.2 Méthodologie générale

La planification de projet nécessite l'acquisition de nombreuses données pour permettre l'ordonnancement. La méthodologie proposée comprend ainsi des activités permettant de déterminer de manière systématique les paramètres de chevauchement utiles à l'accélération du projet. Les paramètres de chevauchement décrivent les opportunités réalistes de chevauchement auxquelles on peut associer des quantités de retouche. Elles seront définies de manière discrète en fonction du type de projet et du suivi qui en est fait. Pour des projets de grande ampleur, les activités sont gérées par période d'une semaine, alors que pour des projets très courts, la journée voire la demi-journée peut être une période acceptable. Les opportunités correspondront à ces périodes afin de pouvoir être appliquées et gérées facilement par les gestionnaires de projet. Les différentes possibilités de chevauchement pour un même couple d'activités sont définies comme des modes de chevauchement.

Les modes de chevauchement sont associés à la relation de précédence possible entre deux activités. Lorsqu'une activité compte plusieurs prédécesseurs, le mode m correspond alors à une combinaison des différentes relations de précédence. Dans la suite du mémoire, on confond les modes de chevauchement avec les modes de précédence. Le premier mode de chevauchement ($m=1$) correspond à une relation de précédence classique avec l'ensemble des prédécesseurs. La Figure 2.1 donne un exemple des différents modes possibles pour une activité avec un prédécesseur.

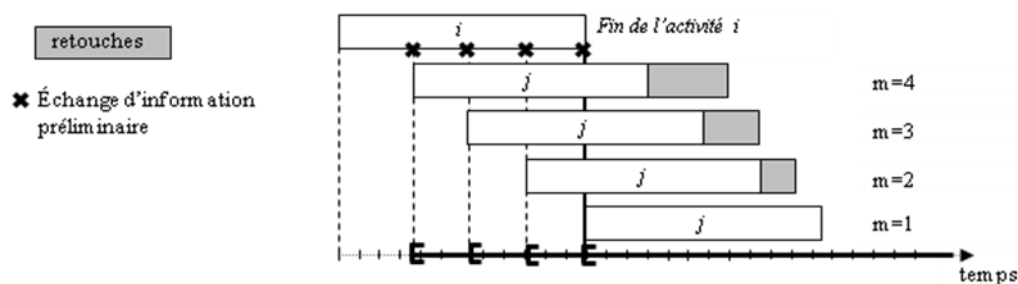


Figure 2.1 : Exemple de différents modes de chevauchement possibles pour un couple d'activités (i, j)

Une fois les différents modes de chevauchement définis, l'ordonnancement du projet permettra d'obtenir un planning de projet accéléré. La Figure 2.2 présente en détail les principales étapes du processus, allant de la décomposition du projet en activités à l'ordonnancement et à l'évaluation de la solution. Ces différentes étapes sont décrites ci-après.

(1) La première étape du processus consiste à décomposer le projet en lots de travail. La taille des lots de travail considérés dépend du niveau de détail exigé pour la bonne gestion du projet. Les lots de travail doivent représenter des activités de complexités et de durées comparables. C'est à partir de ces activités que les possibilités de chevauchement seront établies. Plus le projet est décomposé en activités de courte durée, plus son exécution demande un effort de gestion et de contrôle important. A l'inverse, considérer de nombreuses activités de courte durée permet d'augmenter la possibilité de détecter des opportunités de chevauchement parmi les couples d'activités lors de la suite du processus de planification.

(2) La seconde étape consiste à déterminer la séquence d'activités. La plupart des méthodes de planification de projet décrit les projets sous forme de réseaux où les nœuds représentent les activités (réseau « *Activity on Node* ») ou les relations entre les activités (réseau « *Activity on Arc* »). Ces réseaux décrivent la séquence d'exécution des activités. Cependant, ils ne permettent pas de représenter les interactions entre les activités. Ils ne permettent pas non plus de modéliser les flux d'information entre les activités.

Comme vu précédemment dans la revue de littérature, les matrices de décomposition DSM permettent de modéliser ces relations supplémentaires entre les activités. Les DSM visent à représenter les flux d'information entre les différentes activités et permettent ainsi de repérer les retours d'information. De manière à éviter les retours en arrière, il est ici proposé de

triangulariser les DSM pour obtenir une séquence unidirectionnelle de flux d'information. En dernier recours, il est possible d'agréger les activités ou de les décomposer de manière à éliminer les retours en arrière. Ce processus permet d'obtenir une séquence d'exécution des activités où les relations de précédence et les flux d'information sont orientés dans le même sens.

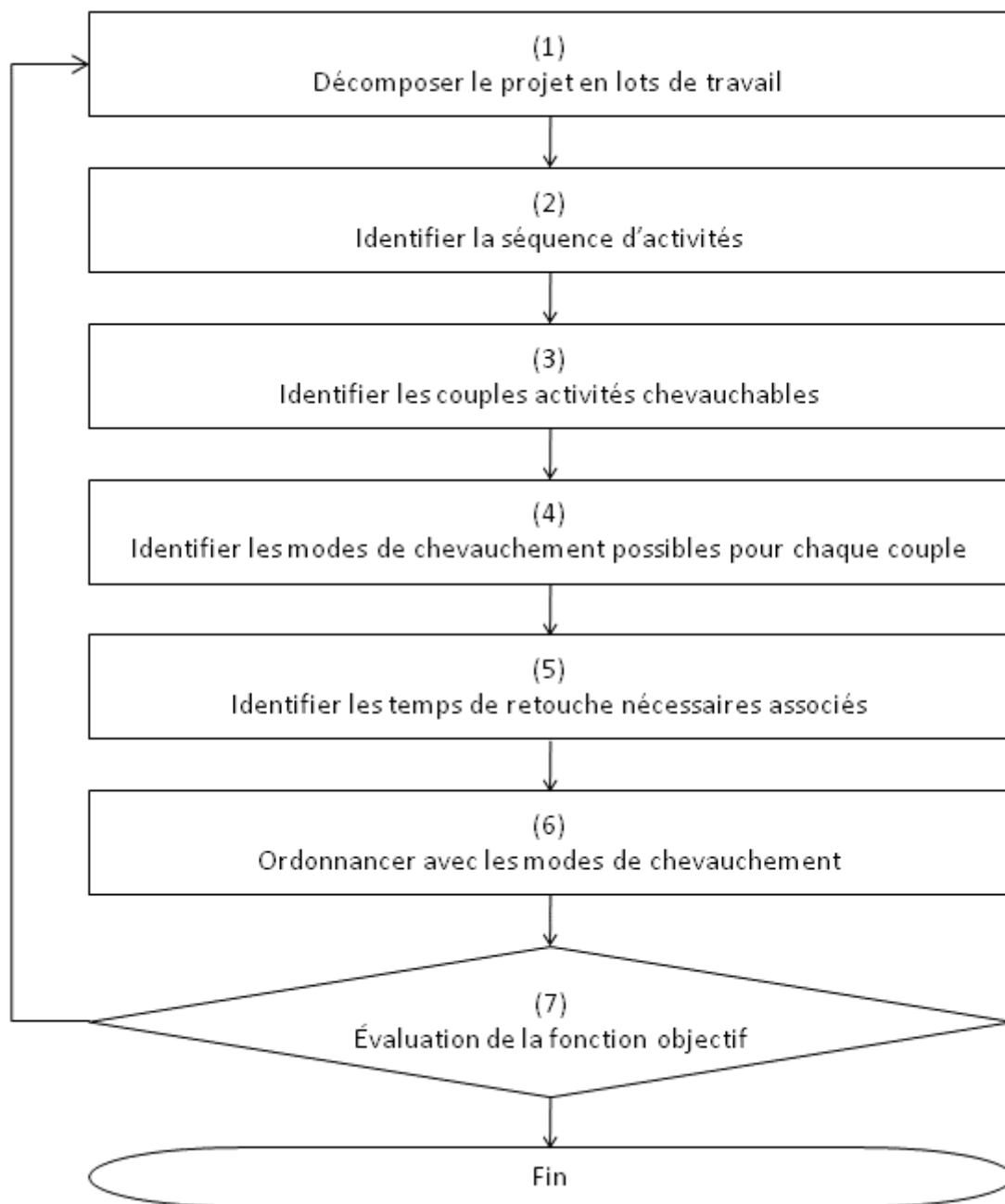


Figure 2.2 : Processus d'accélération de projet avec modes de chevauchement

La méthodologie suppose donc que des études préliminaires ont été menées sur la séquence d'activités de manière à identifier les relations entre les activités et à éliminer les retours en arrière. Les projets considérés par la suite sont composés uniquement d'activités dépendantes et indépendantes et le flux d'information entre activités dépendantes est considéré comme unidirectionnel.

(3) La troisième étape du processus consiste à identifier les couples d'activités chevauchables. L'étude des échanges d'information entre les activités permet de différencier les couples d'activités en deux catégories : les couples chevauchables et les couples non chevauchables. Les premiers représentent les cas où l'activité en aval peut commencer avec des informations préliminaires et recevoir durant son exécution les informations finales de l'activité en amont. De ce fait, on peut chevaucher les deux activités; l'activité en aval peut commencer avant que l'activité en amont ne soit terminée. Les activités non chevauchables sont connectées par des relations de précédence classiques fin-début alors que les activités chevauchables sont connectées par des relations de type fin-début plus un retard, le retard étant le temps de chevauchement.

Les chevauchements sont autorisés lorsque la sensibilité, définie dans le chapitre précédent, sera faible. En effet, pour les couples d'activités dont la sensibilité est élevée, les changements dans l'information délivrée par l'activité en amont impliquent de nombreuses retouches sur l'activité en aval. Le gain associé est donc minime (Krishnan 1996). Dans la pratique, cette règle consiste à autoriser seulement les chevauchements entre des activités d'ingénierie peu sensibles aux changements, ou pour lesquels il existe des solutions qui permettent de limiter les retouches par la suite (surdimensionnement, blocage rapide du *design*, etc.). A l'inverse, on évitera de chevaucher tous les couples d'activités comprenant des activités très sensibles aux changements.

Dans la suite, on note A et P comme l'ensemble des activités chevauchables et non chevauchables respectivement. De même, on note $A(i)$ et $P(i)$ comme l'ensemble des prédécesseurs de l'activité i qui sont respectivement chevauchables et non chevauchables avec l'activité i .

$$Pred(i) = A(i) \cup P(i) \quad \forall i \in S$$

(4) et (5), les quatrième et cinquième étapes consistent à identifier les modes de chevauchement et les durées de retouches associées. La Figure 2.3 représente le mécanisme de chevauchement. L'activité j commence avec des informations préliminaires de l'activité i . Le taux de

chevauchement associé au mode m , α_{ijm} , est exprimé comme une fraction de la durée de l'activité j . Une durée de retouche est par ailleurs ajoutée pour s'accommoder à la mise à jour des informations transmises lors de la réalisation de l'activité j . Le temps d'exécution des deux activités pour le mode de chevauchement m est donc égal à D_{ij} et s'exprime comme suit :

$$D_{ij} = d_i + d_j * (1 - \alpha_{ijm}) + r_{ijm}$$

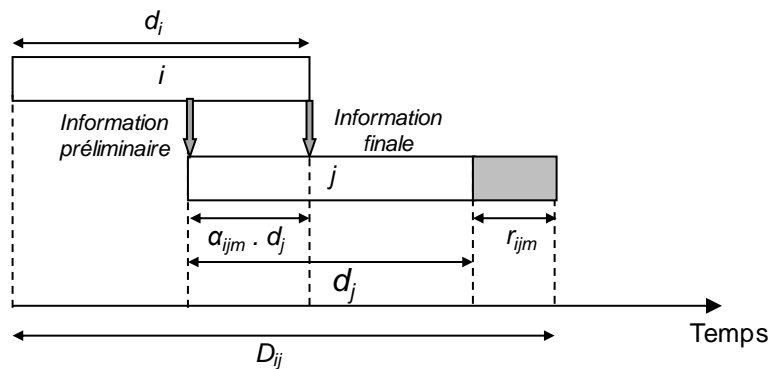


Figure 2.3 : Mécanisme de chevauchement de deux activités i et j dans le mode m

Les pourcentages de chevauchement et les retouches associées aux modes sont définis de manière discrète. En effet, l'ordonnancement d'un projet est établi sur la base de périodes discrètes dans la pratique (heures, jours, semaines, etc.). De plus, l'avancement des activités est mesuré à partir de la réalisation de jalons qui correspondent à des avancements majeurs ou à la réalisation de biens livrables (spécifications, plan, commande de matière première, etc.) définis dans les phases initiales du projet. Les informations préliminaires nécessaires au commencement de l'activité en aval sont issues de la réalisation de ces jalons. Chaque mode de chevauchement est donc caractérisé par un pourcentage de chevauchement, correspondant à une transmission d'information capitale de l'activité en amont, et une quantité de retouche supplémentaire à effectuer sur l'activité en aval.

Les valeurs de α_{ijm} sont déterminées afin de correspondre avec la livraison des biens livrables majeurs de l'activité i . Cette méthodologie permet de s'assurer que les informations préliminaires utilisées sont approuvées. Le problème de la détermination du temps des retouches par rapport au temps de chevauchement est traité dans de nombreux articles de la littérature (Krishnan, Eppinger et al. 1997; Lin, Chai et al. 2009). La définition des modes de chevauchement par rapport aux

jalons permet de recourir facilement aux données historiques, si disponibles, pour estimer les temps de retouches causés par le chevauchement d'activités. Le recours à des estimations obtenues auprès d'experts est aussi possible car la définition des jalons permet de s'assurer des informations transmises lorsqu'ils sont réalisés.

Par la suite, les activités chevauchables peuvent être exécutées de manière conventionnelle sans chevauchement (mode 1) ou de manière accélérée (mode > 1). On suppose qu'il n'y a pas de restriction concernant le nombre de prédécesseurs chevauchables pour une activité. Dans ce cas, on considérera le total de retouches à effectuer comme la somme de retouches associées aux modes de précédences. Suite à l'évaluation des paramètres de chevauchement, l'ordonnancement du projet peut être entrepris. Dans le cadre de ce travail, deux approches d'ordonnancement seront présentées. On aborde dans un premier temps l'ordonnancement à l'aide d'un modèle de programmation linéaire. Connaissant les limites de cette approche exacte, une approche heuristique est présentée dans un deuxième temps.

Une fois l'ordonnancement terminé, l'échéancier obtenu pourra être comparé avec l'échéancier initial, communément appelé *baseline*, en termes de coût et de durée d'exécution. Si le temps d'exécution du projet ainsi que son coût associé sont acceptables, alors l'algorithme s'arrête ici. Le cas échéant, le processus peut être repris à l'étape initiale, en décomposant plus finement les activités pour détecter de nouvelles opportunités de chevauchement, afin d'obtenir un échéancier plus compressé.

L'efficacité du processus d'accélération de projet est tributaire de nombreuses conditions. La qualité des données d'estimation qui évaluent les relations, les échanges d'information, les durées des activités ainsi que leurs consommations en ressource est primordiale. De plus, la considération du chevauchement comme une technique d'accélération de projet nécessite la définition de nombreux paramètres supplémentaires.

2.3 Paramètres nécessaires à la définition des modes de chevauchement

Ainsi, il faut tout d'abord déterminer l'ensemble des activités, leurs consommations de ressources, ainsi que la séquence d'exécution du projet. C'est à partir de cette base que les couples d'activité qui peuvent être chevauchés sont identifiés. Ces couples seront identifiés de manière empirique à l'aide de l'analyse des données d'entrée et de sortie des activités. L'étude

des données historiques de projets réalisés en régime accéléré ou le recours à des planificateurs expérimentés permet aussi d'identifier les activités chevauchables.

Une fois cet ensemble d'activités déterminé, on identifie, pour chaque couple d'activités chevauchables, les différentes opportunités de chevauchement en fonction des données d'entrée nécessaires au commencement de l'activité en aval. Ces données permettent alors de définir les différents modes possibles de précédence.

À chaque mode de précédence, un pourcentage de chevauchement associé, en accord avec la transmission des premières informations nécessaires au commencement de l'activité en aval, est déterminé. Pour chaque mode, il faut associer les retouches liées au chevauchement. La durée des retouches est estimée à l'aide des données historiques sur l'exécution de projet et d'activités similaires.

2.4 Conclusion

Dans cette section, le processus général qui permet d'appliquer notre modèle d'accélération de projet fut présenté. Nous avons ainsi identifié les paramètres indispensables à la bonne exécution du processus. Dans le chapitre suivant, le modèle de base d'ordonnancement qui intègre les opportunités de chevauchement est présenté.

CHAPITRE 3 MODELE OPTIMAL D'ORDONNANCEMENT DE PROJET AVEC CHEVAUCHEMENT D'ACTIVITÉS

3.1 Introduction

Utilisant les paramètres définis précédemment, ce chapitre vise à présenter un modèle de base d'ordonnancement avec chevauchement d'activités. Les hypothèses et notations utilisées de ce modèle sont d'ailleurs présentées à la section suivante. Une étude permettant d'évaluer la pertinence des paramètres utilisés et de la modélisation proposée sera par la suite présentée. Finalement, les limitations d'un tel modèle d'ordonnancement seront par la suite discutées avant de présenter une extension.

3.2 Hypothèses et notations

Les notations utilisées dans la suite de nos travaux sont décrites dans le Tableau 3.1.

Tableau 3.1 : Notations utilisées

Symboles	Définition
S	Ensemble des activités
n	Nombre d'activités réelles
d_j	Durée d'exécution de l'activité j
A	Ensemble des couples d'activités chevauchables
P	Ensemble des couples d'activités non chevauchables
$A(j)$	Ensemble des prédécesseurs de j qui sont chevauchables avec l'activité j
$P(j)$	Ensemble des prédécesseurs de j qui ne sont pas chevauchables avec l'activité j
$Pred(j) = A(j) \cup P(j) \quad \forall j \in S$	Ensemble des prédécesseurs de j
R	Ensemble des ressources
R_k	Quantité disponible constante pour la ressource k
R_{jk}	Nombre d'unités de ressource k nécessaires par période pour l'exécution normale de l'activité j
θ_k	Coût horaire de la ressource k
m_j	Nombre de modes d'exécution possibles pour l'activité j
α_{ijm}	Pourcentage de la durée de chevauchement entre les activités i et j dans le mode d'exécution m , exprimé comme une fraction de d_j

r_{jm}	Durée des retouches associées au mode d'exécution m pour l'activité j
T	Durée maximale d'exécution du projet
$t = 0, \dots, T$	Périodes
EF_j	Date de fin au plus tôt de l'activité j
LF_j	Date de fin au plus tard de l'activité j

Comme expliqué précédemment, l'identification des activités chevauchables est une étape clé pour la réussite du processus d'accélération. On suppose que lors de l'ordonnancement du projet, le réseau d'activités, les activités, les consommations de ressources et les opportunités de chevauchement sont connues et fixes. On suppose aussi qu'aucune de ces données d'entrée n'est modifiable lors de l'exécution du projet. Ceci constitue notre première hypothèse.

La seconde hypothèse est issue de l'application du processus et de l'utilisation des DSM dans la génération de la séquence d'activité. On suppose que la séquence d'activité obtenue permet des échanges d'information unidirectionnels entre les activités. Cela résulte de la seconde étape du processus présenté précédemment, qui correspond à la réalisation de la séquence d'activités.

Aussi, le chevauchement d'activités implique des réunions de coordination et des retouches supplémentaires. Ces activités supplémentaires impliquent deux types de coûts additionnels. Les premiers, et les plus importants, sont les coûts liés aux retouches. On suppose que les retouches requièrent les mêmes ressources que l'activité elle-même. Le coût associé à une retouche est donc le coût d'utilisation des ressources pour sa durée.

Les deuxièmes coûts associés au chevauchement d'activités sont les coûts de communication. L'exécution des activités en parallèle nécessitent un effort supplémentaire de coordination et de communication entre les ressources. On suppose comme troisième hypothèse que ces activités de coordination sont d'une durée nulle, tout comme les coûts associés. On justifie cette hypothèse par le caractère négligeable de la durée des réunions de coordination par rapport à la durée des activités elles-mêmes. Aussi l'utilisation de technologies pour la communication (internet, téléphone) requiert un coût largement négligeable vis-à-vis du coût d'exécution d'un projet.

Enfin pour l'application du modèle d'ordonnancement, on suppose que tous les paramètres de chevauchement, modes et temps de retouche associés, sont préalablement connus, et que, comme pour les couples d'activités chevauchables, on ne peut agir sur ces paramètres.

Une fois ces hypothèses posées, le problème d'ordonnancement peut être abordé selon plusieurs objectifs. L'approche classique des modèles RCPSP consiste à minimiser le temps d'exécution du projet tout en respectant les contraintes de capacité des ressources. Dans ce chapitre, un modèle optimal d'ordonnancement de projet avec chevauchement d'activités sera présenté. Le générateur de projets, utilisé pour tester ce modèle, est présenté à la section suivante. La génération de nombreuses instances de projet permettra notamment de mesurer l'influence des paramètres caractéristiques du projet sur l'efficacité du chevauchement d'activités comme technique d'accélération de projet, ainsi que les limites associées à une telle modélisation.

3.3 Le générateur de projet

Afin de tester trois modèles d'ordonnancement de projet avec modes de chevauchement, un modèle de base puis deux extensions, il est nécessaire dans un premier temps de générer des instances de projets. Dans le but d'atteindre nos objectifs expérimentaux, le générateur de projets PROGEN développé par Kolisch et Sprecher (Kolisch and Sprecher 1997) est ici utilisé pour générer des réseaux d'activités, auxquels des paramètres de chevauchement ont été ajoutés. Les sections suivantes indiquent comment ont été générées les différentes données de projets.

3.3.1 Génération du réseau de projet avec contraintes de ressources

Le générateur de projet PROGEN permet de générer des réseaux d'activités et des contraintes de ressources qui respectent des paramètres prédéfinis. Dans cette étude, des réseaux de 32 activités, dont deux activités fictives qui sont les activités de début et de fin de projet ont été générés. Chaque activité consomme des ressources parmi les quatre types de ressources définies, chacune représentant des compétences distinctes. A chaque compétence est associé un nombre de ressources disponibles. Les paramètres utilisés pour générer les projets sont C , RF et RS . Ceux-ci sont décrits ci-après. Pour les autres paramètres utilisés par PROGEN, les paramètres définis par défaut ont été conservés.

Le paramètre C , pour « *Network Complexity* », représente le nombre moyen de relations entre les activités. Il définit la complexité du réseau d'activités. Dans la bibliothèque de projet développée par les deux auteurs, le paramètre C varie entre les valeurs de 1,5 et 2,1. Plus la valeur de C est importante, plus le problème d'ordonnancement sera facile à résoudre car il est plus contraint, donc il y a moins de possibilités d'ordonnancement à explorer.

Le paramètre RF , ou « *Resource Factor* », définit, en pourcentage, le nombre moyen de compétences nécessaires à l'exécution normale de l'activité. Une valeur égale à 1 signifie que toutes les activités nécessitent les quatre compétences pour leur exécution. Une valeur égale à 0,5 signifie qu'en moyenne les activités utilisent deux compétences distinctes pour leur réalisation.

Enfin, le paramètre RS , ou « *Resource Strength* », est utilisé pour déterminer le nombre de ressources disponibles par compétence. Il définit la sévérité des contraintes de ressources. En effet, pour chaque type de ressources, on détermine le nombre disponible de la manière suivante :

$$R_k = Q_{k,\min} + Round(RS * (Q_{k,\max} - Q_{k,\min}))$$

Où $Q_{k,\min}$ est la quantité minimum acceptable pour la ressource k , égale à la demande maximale parmi les activités pour la ressource k . $Q_{k,\max}$ est égale à la demande maximale observée pour la ressource k lorsque l'on exécute le projet sans considérer les contraintes de ressources. RS est le paramètre qui a le plus d'influence sur le temps de résolution du problème RCPSP classique. Plus RS est petit, plus le problème est complexe à résoudre. Des études (Kolisch et al., 1995) montrent qu'il y a un facteur 1000 entre $RS=1$ et $RS=0,2$ lorsque l'on considère le temps de résolution.

Ces trois paramètres permettent de générer, par l'utilisation de PROGEN, les réseaux d'activités avec les durées associées à l'exécution de chaque activité, les relations entre les activités et les ressources nécessaires. Une fois les réseaux d'activités et les contraintes de ressources générées, il faut maintenant définir l'ensemble des paramètres nécessaires à la prise en compte des opportunités de chevauchement. Dans la prochaine section, nous verrons comment nous avons évalué celles-ci.

3.3.2 Paramètres de chevauchement

Les paramètres de chevauchement nécessaires à l'exécution de notre processus d'accélération de projet sous contrainte de ressources avec mode de chevauchement sont les couples d'activités chevauchables, les modes de chevauchement, les taux de chevauchement et les temps de retouche associés (A , m_j , α_{ijm} et r_{jm}). On choisit de déterminer aléatoirement les possibilités de chevauchement à partir de trois facteurs représentatifs des caractéristiques du projet :

- R définit le pourcentage de couples d'activités chevauchables parmi les couples d'activités. Ce paramètre décrit les opportunités de chevauchement et la stratégie

d'exécution du projet. Il reflète les résultats de l'analyse des relations et des flux d'information entre activités qui permet de déceler un nombre plus ou moins d'opportunités de chevauchement. Dans la réalité, les couples chevauchables seront établis au cas par cas en considérant les flux d'information et les dépendances entre activités.

- C_{max} définit le pourcentage maximum de chevauchement d'un couple d'activités. Ce paramètre reflète, pour les couples d'activités chevauchables, les opportunités, plus ou moins grandes de chevauchement. Il correspond au pourcentage associé au mode de chevauchement le plus agressif. Pour nos cas d'étude, on considère que tous les couples d'activités chevauchables ont la même valeur de C_{max} . Dans la réalité, cette valeur sera déterminée au cas par cas, pour chaque couple d'activités.
- B décrit la relation entre le temps de chevauchement et le temps de retouche. On considère ici une relation linéaire entre le temps de chevauchement et le temps de retouche. Cette hypothèse permet de générer facilement des cas d'étude. Dans la réalité, les temps de retouches associés à chaque mode seront évalués séparément.

A l'aide de ces paramètres, il est maintenant possible de générer l'ensemble des données nécessaires à l'application du processus d'accélération de projet avec contraintes de ressources et modes de chevauchement. J'ai ainsi développé un générateur de données de chevauchement qui pourra être utilisé pour générer aléatoirement des données et tester ainsi les performances d'algorithmes d'ordonnancement de projet avec contraintes de ressources et modes de chevauchement. Le générateur est implémenté à l'aide d'une macro en Visual Basic dans Excel 2007, dont le fonctionnement général est présenté dans la Figure 3.1.

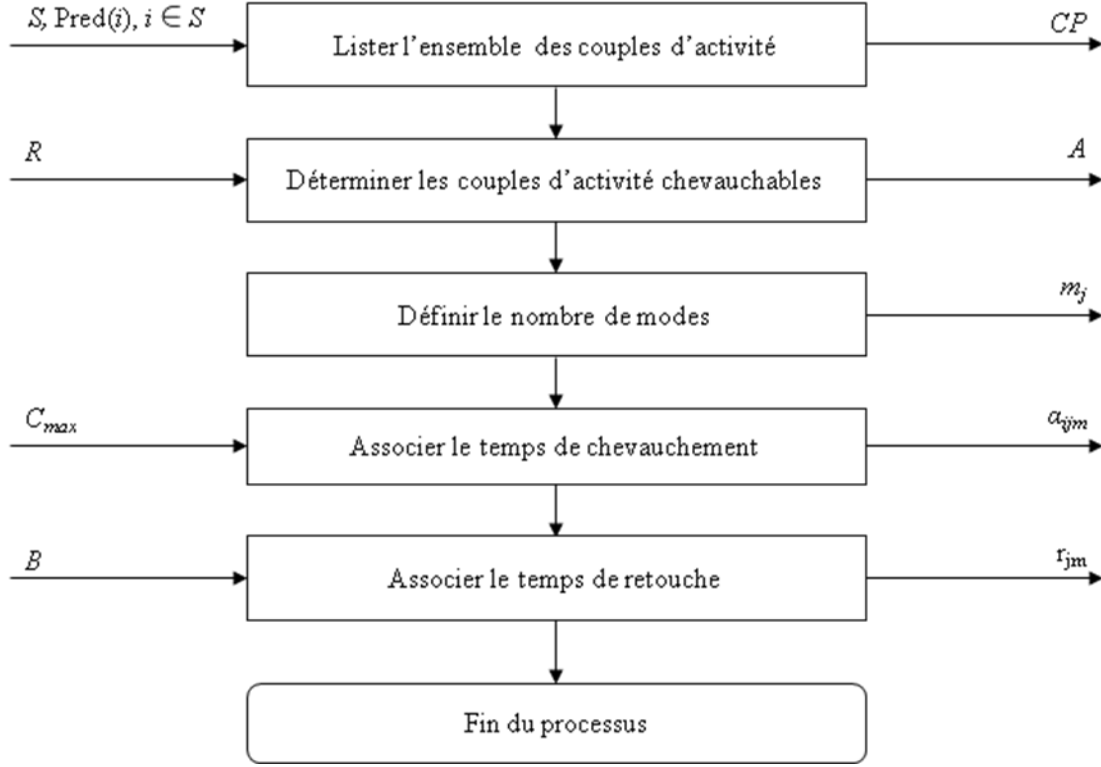


Figure 3.1 : Génération des données de chevauchement

La Figure 3.1 montre le fonctionnement global de la macro de génération des données de chevauchement. L'ensemble des couples d'activités est noté CP . Les couples d'activités chevauchables sont déterminés aléatoirement de manière à respecter la contrainte fixée par le paramètre R :

$$\text{card}(CP) * R = \text{card}(A)$$

Une fois les couples d'activités chevauchables identifiés, on leur associe un nombre de modes prédéfinis. Dans notre cas, on associe à chaque couple, quatre modes de précédence différents. Ensuite, pour chaque couple d'activités, on définit les différents pourcentages de chevauchement associés aux modes à l'aide du paramètre d'entrée C_{max} .

$$\alpha_{ijm} = \text{round}\left(\frac{C_{max} * (m-1)}{3} * d_i\right) \quad , \forall j \in A, (i, j) \in CP, \forall m \in [1, m_j]$$

De la même manière, on détermine les temps de retouches associés aux modes, à l'aide du paramètre d'entrée B , qui définit le temps de retouches nécessaire pour s'adapter à l'évolution de l'information. On considère que si une activité chevauche plusieurs prédécesseurs, alors le temps de retouche total sera la somme des retouches associées au chevauchement avec chacun d'entre eux.

$$r_{jm} = \sum_{i \in \text{Pred}(j)} \text{round}(B * \alpha_{ijm} * d_i) \quad , \forall j \in A, \forall m \in [1, m_j]$$

Finalement, le générateur permet d'obtenir l'ensemble des données nécessaires à l'ordonnancement d'un projet avec chevauchement d'activités. Ces données peuvent être facilement configurées à l'aide des trois paramètres présentés.

On rappelle que la génération aléatoire de telles données de projet permettra de montrer la validité de la méthode d'ordonnancement, ainsi que la capacité des paramètres définis à modéliser fidèlement l'impact du chevauchement d'activités sur les coûts et les délais reliés à l'exécution du projet.

3.4 Modèle de base d'ordonnancement

Le scénario classique d'ordonnancement de projet consiste à minimiser la durée d'exécution du projet. Une fenêtre de temps est déterminée $[EF_i, LF_i]$ pour l'exécution de chaque activité i afin de réduire le nombre de variables. Elles peuvent être déterminées par les praticiens en fonction des exigences intrinsèques au projet, et à l'aide d'un algorithme de passe avant et passe arrière considérant que le projet doit commencer au temps 0 et en considérant un temps T comme une borne supérieure du temps d'exécution.

Les variables de décision du problème sont les suivantes :

$$X_{jtm} = \begin{cases} 1 & \text{si l'activité } j \text{ est exécutée dans le mode } m \text{ et finie au temps } t \\ 0 & \text{sinon} \end{cases}$$

$$\forall j \in S, \forall t \in [0, T], \forall m \in [1, m]$$

Le mode associé à une activité correspond à un mode de précédence. La décision sur les modes de précédence peut être définie par trois cas différents. Si le couple d'activités (i, j) n'est pas chevauchable, alors la décision est de ne pas chevaucher, et on applique le seul mode possible de

précédence. Si le couple (i,j) est un couple d'activités chevauchables, alors, on peut choisir de chevaucher les deux activités ($m > 1$) ou alors de les exécuter en série ($m=1$). Le problème d'ordonnancement avec contraintes de ressources et modes de chevauchement peut être formulé sous la forme d'un modèle de programmation linéaire en nombres entiers présenté ici :

$$\text{Minimiser } \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} \quad (1)$$

Contraintes :

$$\sum_{m=1}^{m_j} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} (t - d_j \cdot (1 - \alpha_{ijm}) - r_{jm}) \cdot X_{jtm} \quad \forall j \in S \quad \forall i \in \text{Pred}(j) \quad (2)$$

$$\sum_{j=2}^n \left[R_{jk} \cdot \left(\sum_{m=1}^{m_j} \sum_{b=t}^{t+d_j-1+r_{jm}} X_{jbm} \right) \right] \leq R_k, \quad \forall k \in R, \quad \forall t \in [0, T] \quad (3)$$

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} t \cdot X_{jtm}, \quad \forall i \in A(j) \quad (4)$$

$$\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} X_{jtm} = 1 \quad \forall j \in S \quad (5)$$

$$X_{jtm} = \{0,1\}, \quad \forall j \in S, \quad \forall t \in [0, T] \quad (6)$$

L'objectif (1) minimise la date de fin de l'activité fictive de fin du projet. Il permet donc de minimiser la durée d'exécution du projet en assurant que toutes les activités ont été réalisées. La contrainte (2) prend en compte les différents modes de précédence applicables et assure le respect des contraintes de précédence en prenant en compte les possibilités de chevauchement. La contrainte (3) définit les contraintes de ressources. La contrainte (4) permet de s'assurer que l'activité en aval ne puisse se terminer avant son prédécesseur, même en cas de chevauchement. On a supposé préalablement que toute activité à besoin des informations finales de son prédécesseur pour se terminer. La contrainte (5) s'assure qu'un seul mode de précédence est appliqué pour chaque activité et que chaque activité est terminée à la fin du projet. Enfin, la contrainte (6) définit la variable de décision binaire.

Cette modélisation du problème permet d'obtenir un planning de durée minimale pour l'exécution du projet, tout en prenant en compte les possibilités de chevauchement et les temps de retouche associés.

Une fois la solution déterminée, le coût associé à la stratégie d'exécution est calculé de la manière suivante :

$$Coût = \sum_{j \in S} \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} \sum_{k \in R} X_{jtm} * R_{jk} * r_{jm} * \theta_k$$

Où θ_k est le coût horaire de la ressource k .

La section suivante présente les résultats du modèle d'ordonnancement proposé sur de nombreux exemples. L'objectif est d'évaluer l'efficacité du chevauchement en termes de coût additionnel et de réduction du temps d'exécution.

3.5 Influence des caractéristiques du projet sur l'efficacité du chevauchement

Cette section reprend en grande parties les données et résultats présentés dans l'article « *Evaluating the effectiveness of task overlapping as a risk response strategy in engineering projects* » soumis pour publication dans un numéro spécial de la revue *International Journal of Project Organisation and Management*. Cet article est présenté en intégralité en Annexe 1.

3.5.1 Données expérimentales

De manière à démontrer l'importance des paramètres intrinsèques au projet et l'importance de prendre en compte les contraintes de ressources quand on réalise un planning en régime accéléré avec le chevauchement d'activités, j'ai testé le modèle présenté avec différentes instances de projets présentant les mêmes caractéristiques.

On réalise un plan d'expérience afin de tester l'impact du chevauchement. Les valeurs présentées dans le Tableau 3.2 sont les valeurs fixes pour l'ensemble de notre plan d'expérience. Je justifie ce choix par le fait que les paramètres C et RF sont des paramètres internes à la nature du projet et ne peuvent en aucun cas être modifiés par les équipes de projet. La complexité des relations entre les activités (C) et l'implication des ressources dans les activités (RF) sont donc deux paramètres qui ne peuvent être modifiés une fois l'envergure du projet définie. Aussi le paramètre B qui définit le pourcentage de retouches nécessaires suite au chevauchement est considéré comme un paramètre constant dans notre plan d'expérience. L'importance des retouches dépend

en grande partie de la nature du projet. Les valeurs choisies correspondent toutes à des valeurs médianes des valeurs utilisées dans les articles de la littérature.

Tableau 3.2 : Valeurs fixes dans le plan d'expérience

C	2,1
RF	0,5
B	0,4

Le Tableau 3.3 représente les différentes valeurs utilisées dans notre plan d'expérience, dans lequel trois paramètres sont variables : RS (« *ressource strength* »), R (pourcentage de couple d'activité chevauchables) et C_{max} (pourcentage maximum de chevauchement).

Tableau 3.3 : Différentes valeurs des paramètres du plan d'expérience

RS	0,50	0,75	1,0
R	0,20	0,40	0,60
C_{max}	0,25	0,50	0,75

On génère tout d'abord 9 réseaux de projet avec les caractéristiques de C et RF . Pour chaque projet on génère 3 instances de projets avec les différentes caractéristiques de R et avec C_{max} égal à sa valeur maximale, 0,75. On fixe le nombre de modes à 4, et on les définit comme illustré sur la Figure 3.2. On obtient alors 27 instances de projet. Les couples d'activités chevauchables pour R égal à 0,50 et 0,25 sont sélectionnés à partir de la liste générée aléatoirement avec R égal à 0,75 et 0,50 respectivement. Ainsi on s'assure que les opportunités de chevauchement d'une instance avec R égal à 0,25 sont comprises dans les opportunités d'une instance avec R égal à 0,50 et 0,75. Cette caractéristique rend les instances comparables entre elles, lorsque l'on fixe tous les autres facteurs et que l'on fait varier R uniquement. On obtient 81 projets.

Pour les instances avec C_{max} égal à 0,50 et 0,25, on garde les mêmes données de chevauchement, en limitant le nombre de modes à 3 et 2 respectivement. Entre deux instances ayant les mêmes

caractéristiques et deux valeurs de C_{max} différentes, les opportunités de chevauchement sont les mêmes, seul le mode maximum autorisé de chevauchement change. Cette caractéristique rend les instances comparables entre elles, de la même manière que pour R.

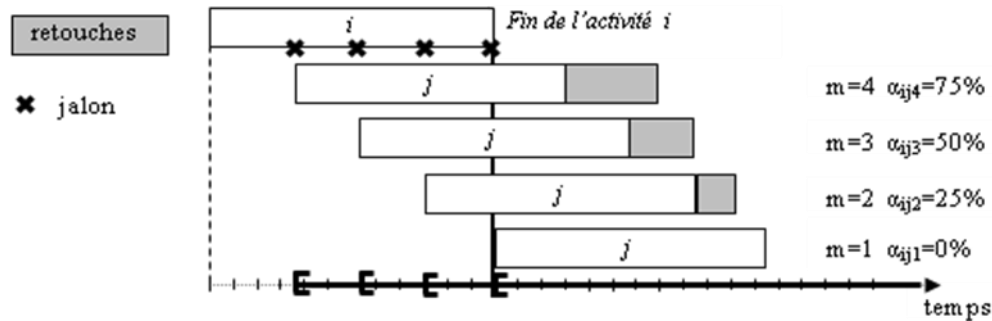


Figure 3.2 : Modes de chevauchement pour $C_{max}=0,75$

Finalement on dispose de 243 projets comparables, sur lesquels on étudiera l'efficacité du chevauchement à accélérer l'exécution d'un projet et son impact sur le coût du projet. Les caractéristiques des instances générées permettront d'analyser facilement les impacts indépendants de chaque facteur variable du plan d'expérience sur les résultats.

3.5.2 Analyse des résultats

Les 243 projets générés ont été implantés dans le logiciel d'optimisation AMPL Studio v1.6.j. J'ai résolu le problème d'ordonnancement avec chevauchement d'activités avec le solveur CPLEX 12.2, sur un ordinateur personnel (2,22GHz, 3,00Go de RAM). Les résultats de l'ordonnancement seront comparés à ceux d'un ordonnancement standard avec contraintes de ressources mais sans chevauchement d'activités. L'efficacité du chevauchement comme technique d'accélération de projet sera donc analysée en comparant le temps d'exécution du projet en régime normal, sans chevauchement, et en régime accéléré, avec chevauchement d'activités. La réduction sera présentée en pourcentage du temps d'exécution du projet. Les coûts additionnels issus des retouches seront aussi analysés de la même façon. Pour y arriver, le coût d'utilisation d'une unité de ressource est fixé à 100\$ par période. Le coût associé au projet, calculé en faisant la somme des coûts d'utilisation des ressources et des coûts fixes. Les coûts fixes sont évalués arbitrairement à 20,000\$.

L'influence des différents paramètres du plan d'expérience est analysée à travers une analyse de la variance. Le diagramme de Pareto est présenté dans la Figure 3.3. Il montre que les trois

facteurs utilisés permettent d'expliquer 80% de la variation de l'efficacité du chevauchement en termes de réduction du temps d'exécution du projet. Les facteurs choisis ont donc un impact significatif sur l'efficacité du chevauchement, et cela me conforte dans le choix de ces paramètres pour décrire les opportunités de chevauchement d'activités et les caractéristiques globales du projet.

Dans ce diagramme de Pareto, on observe que les trois paramètres de notre plan d'expérience ont une influence significative sur la réduction du temps d'exécution du projet, notamment le paramètre RS qui décrit la sévérité des contraintes de ressources. Cela indique clairement qu'il faut prendre en compte les contraintes de ressources lors de l'étude d'un processus d'accélération de projet avec chevauchement d'activités.

Le paramètre R , qui décrit la proportion de couples d'activités qui sont chevauchables, est le paramètre le plus influent vis-à-vis de la réduction du temps d'exécution du projet. Ceci confirme qu'un effort important doit être fait sur la définition des couples d'activités chevauchables.

De plus, l'influence significative de C_{max} donne une information importante sur l'effort à fournir pour déterminer les différents modes de précedence entre les activités, et de ce fait, le mode de chevauchement maximum.

On observe aussi qu'il existe de fortes interactions entre les paramètres qui décrivent le projet, notamment entre les contraintes de ressources et le nombre d'opportunités de chevauchement, décrit par R , et le pourcentage maximum de chevauchement, décrit par C_{max} . Cette remarque ne fait qu'appuyer l'importance de l'impact des contraintes de ressources.

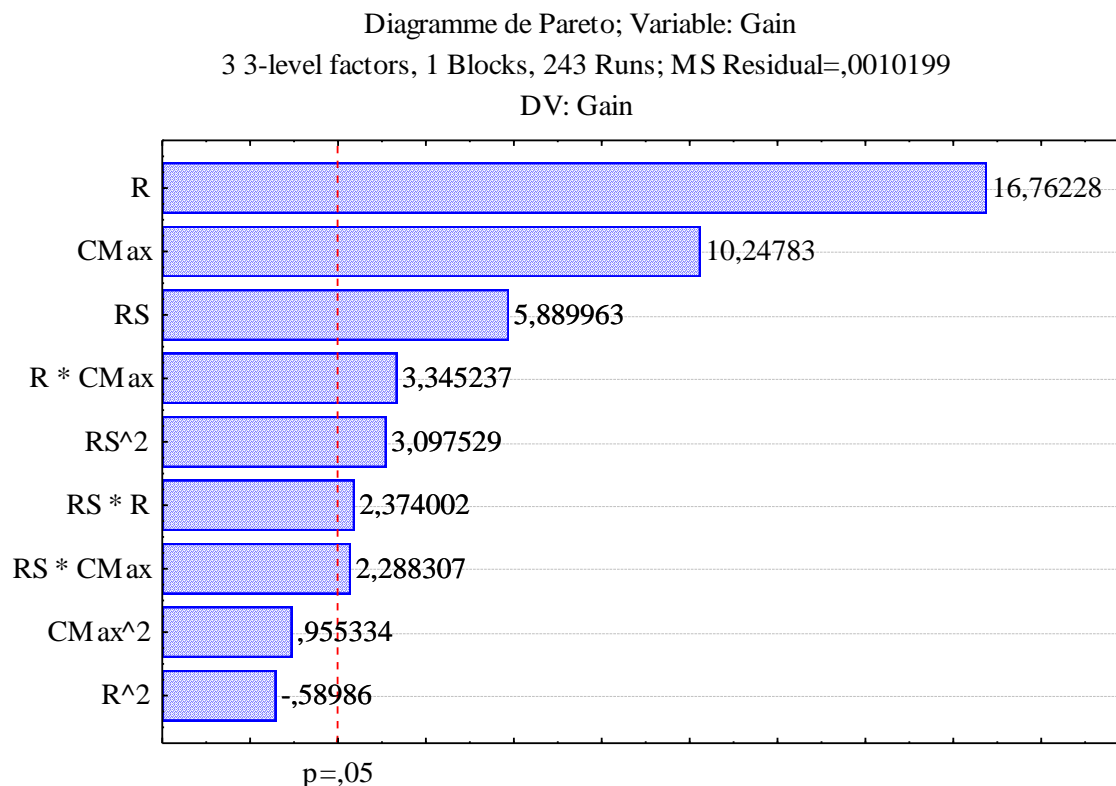


Figure 3.3 : Diagramme de Pareto

La Figure 3.4 illustre l'impact du facteur *RS*, qui définit la sévérité des contraintes de ressources, sur l'efficacité du chevauchement pour réduire le temps d'exécution d'un projet. On peut observer que *RS* a un effet important sur la durée d'exécution du projet, mais n'a pas d'impact significatif en termes de coûts associés (Figure 3.5). En effet, la réduction de la valeur de *RS* diminue les opportunités de chevauchement et le gain de temps associé. Cela est dû au fait que durant le temps de chevauchement entre deux activités, les ressources utilisées sont les ressources nécessaires pour l'exécution des deux activités. Il faudra donc assurer la disponibilité de ces ressources, ce qui réduit donc les possibilités de chevauchement lorsque les contraintes sont fortes. Aussi, le chevauchement d'activités entraîne des retouches dans l'activité en aval, dues à l'acquisition de nouvelles données lors de son exécution. Ces retouches induisent une charge de travail supplémentaire et augmentent ainsi la consommation des ressources. De fortes contraintes de ressources limitent énormément les possibilités de chevauchement. L'écart important entre $RS=0,75$ et $RS=0,5$, où la réduction du temps d'exécution du projet passe de 8,8% à 5,8% montre l'importance de prendre en compte les contraintes de ressources lors de la planification de projet.

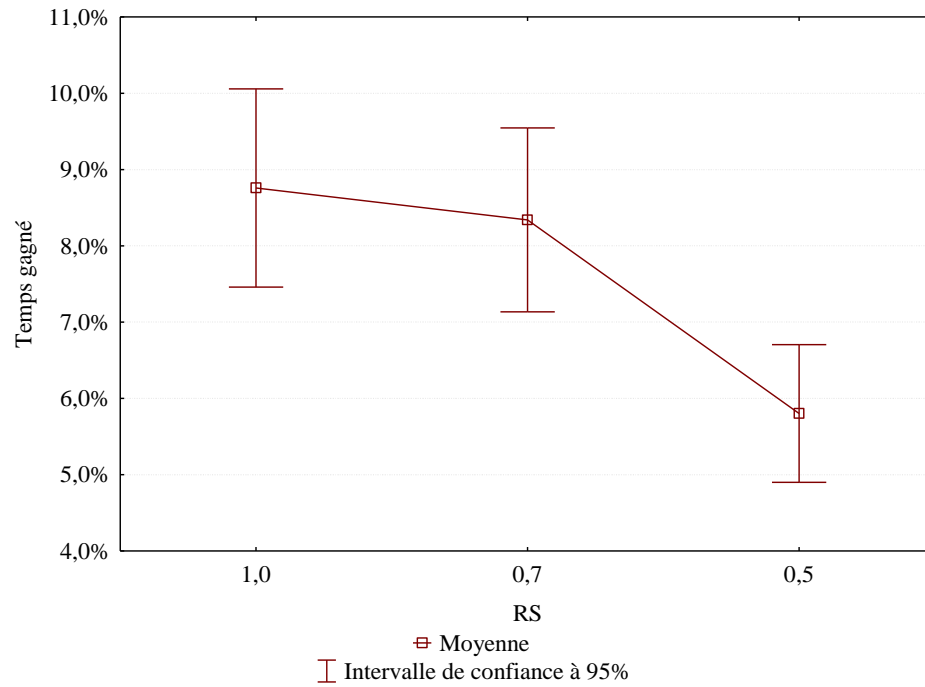


Figure 3.4 : Impacts de RS sur le temps gagné

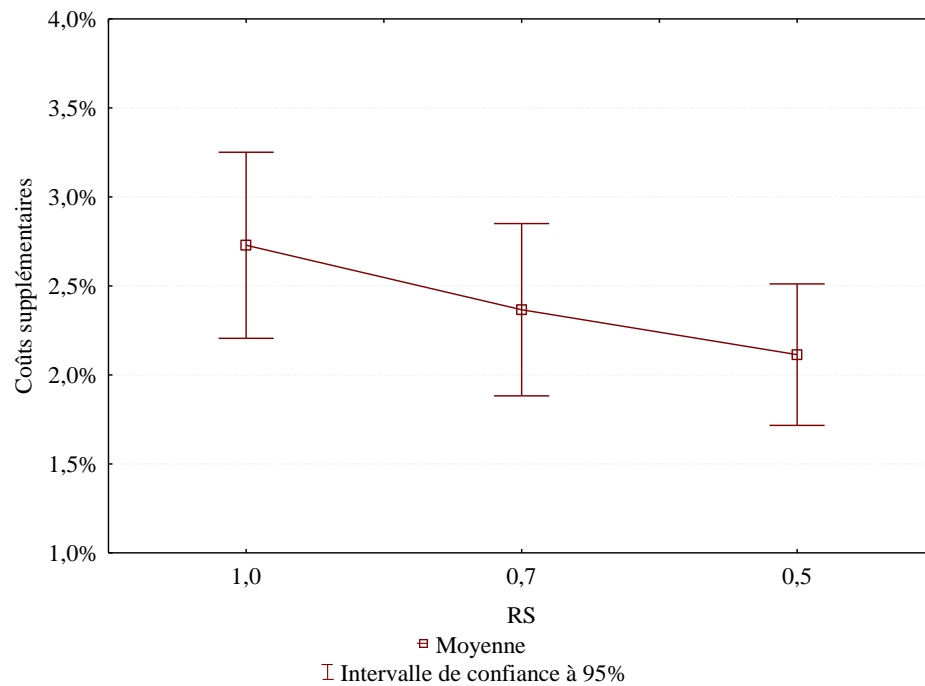


Figure 3.5 : Impacts de RS sur les coûts supplémentaires

L'étude du second paramètre R , qui décrit le pourcentage de couples d'activités chevauchables au sein du projet est représenté dans la Figure 3.6. Lorsque R augmente, on observe que le

chevauchement permet de réduire plus efficacement la durée d'exécution du projet, passant de 3,5% à 12% lorsque R passe de 20% à 60%. De nombreuses opportunités de chevauchement permettent de passer outre les contraintes de ressources et donc de réduire de façon plus importante le temps d'exécution du projet. L'impact sur le coût (Figure 3.7) est concave : plus le nombre de couples d'activités chevauchables est important, moins l'impact sur le coût d'exécution du projet est significatif. On explique cela par le fait que lorsque de nombreux couples sont chevauchables, les modes choisis lors de l'ordonnancement sont plus souvent des modes qui impliquent peu de chevauchement, donc peu de retouches, et par conséquent un coût supplémentaire associé moins important. Ce point met en avant encore une fois, l'importance d'identifier correctement les couples d'activités chevauchables lors de la troisième étape du processus d'ordonnancement avec chevauchement d'activités.

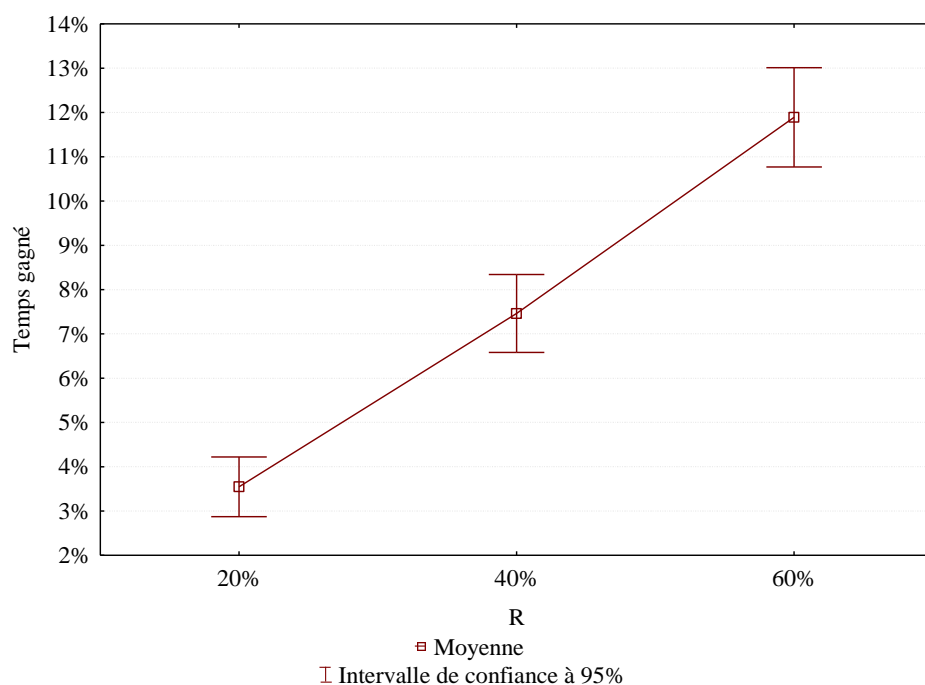


Figure 3.6 : Impacts de R sur le temps gagné

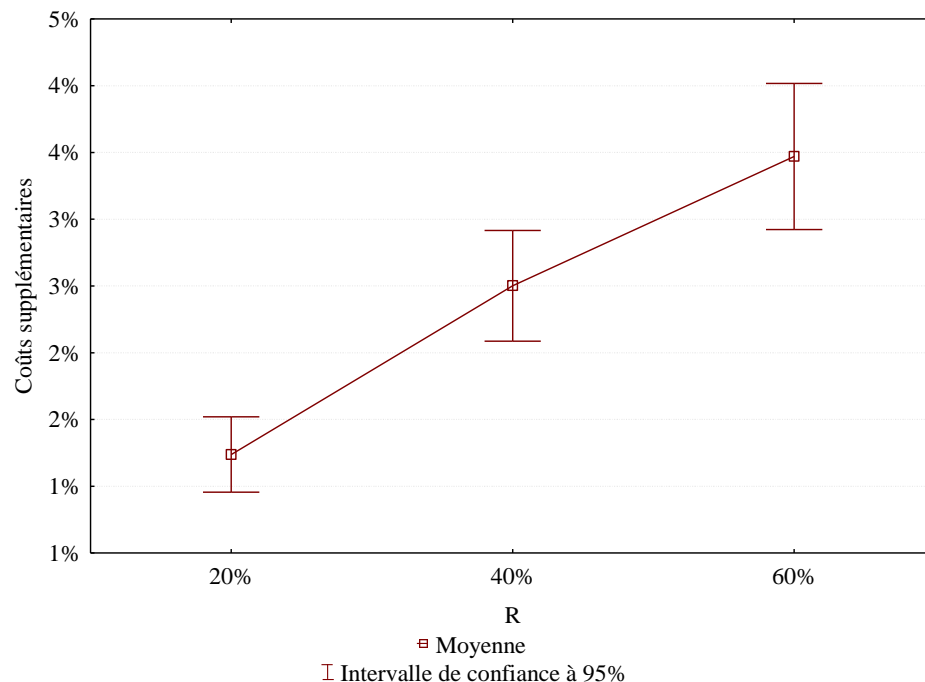


Figure 3.7 : Impacts de R sur les coûts supplémentaires

Enfin, la Figure 3.8 montre l'impact du paramètre C_{max} sur le temps d'exécution du projet. L'augmentation du pourcentage maximum de chevauchement provoque une augmentation concave de la réduction du temps d'exécution du projet. Des modes de chevauchement autorisant un chevauchement important permettent de réduire plus efficacement la durée du projet, mais les contraintes de ressources et les retouches associées au chevauchement tendent à réduire les possibilités de réaliser les activités de manière simultanée. L'impact au niveau des coûts est linéaire (Figure 3.9) ce qui est en accord avec la relation entre le temps de chevauchement et les retouches associées, le coût étant directement calculé à partir des retouches.

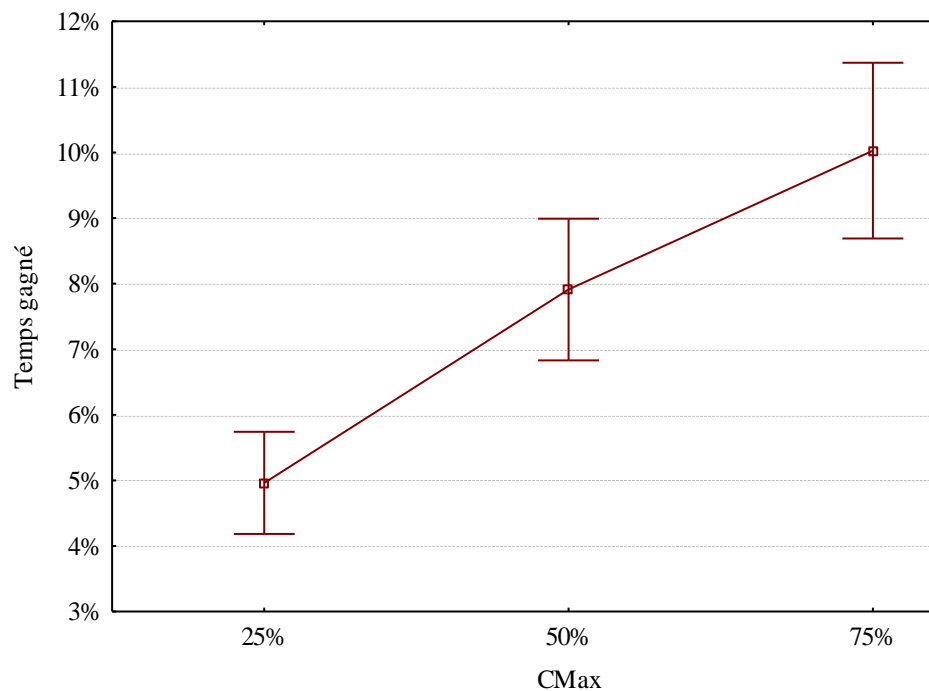


Figure 3.8 : Impacts de C_{\max} sur le temps gagné

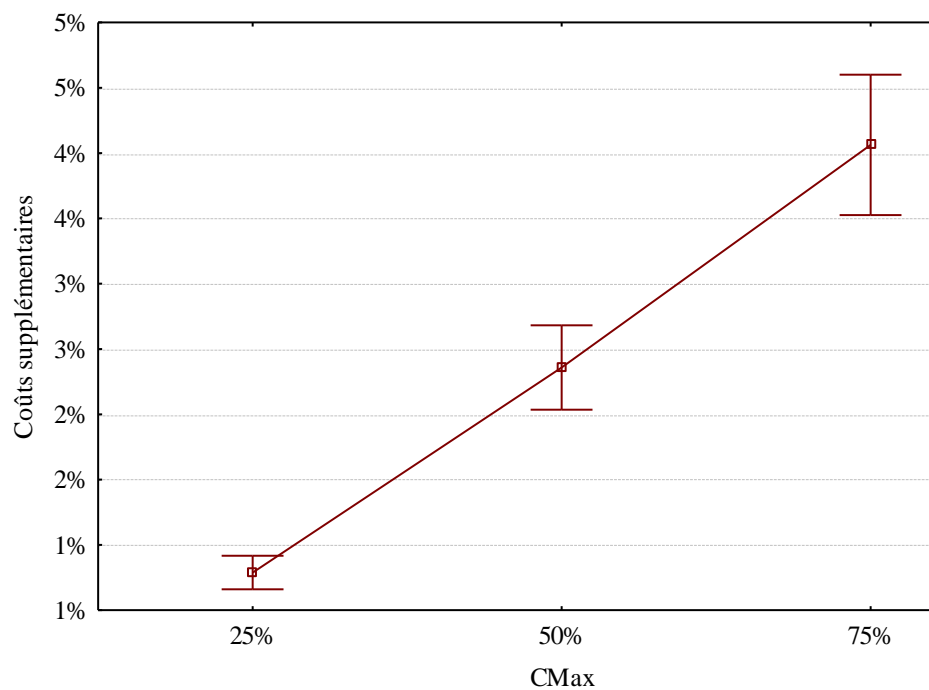


Figure 3.9 : Impacts de C_{\max} sur les coûts supplémentaires

Ces résultats expérimentaux permettent de montrer que de faibles contraintes de ressources et de nombreuses opportunités de chevauchement, associées à un important pourcentage de

chevauchement acceptable sont les conditions favorables à l'application du chevauchement comme une technique d'accélération de projet. Le facteur le plus important est, nous l'avons vu sur le diagramme de Pareto, le nombre de couples d'activités éligibles au chevauchement. Le facteur RS , qui définit la sévérité des contraintes de ressources est le facteur qui a le moins d'impact direct sur l'efficacité du chevauchement. Cependant, nous avons pu observer dans le diagramme de Pareto présenté précédemment que le facteur RS est à l'origine de nombreuses interactions avec les autres facteurs étudiés C_{max} et R , ce qui rend son influence non négligeable.

3.6 Limitations

Cette section présente les limitations rencontrées lors de l'application du modèle de base proposé.

3.6.1 Considérations des coûts associés au chevauchement d'activités

Tout d'abord, le modèle proposé dans ce chapitre considère les coûts mais ne les traite pas de façon optimale. En effet, le modèle d'optimisation proposé permet de trouver une solution à durée d'exécution minimale. Cependant, il y a théoriquement plusieurs stratégies de chevauchement qui permettent l'exécution du projet en un temps minimal, et le modèle utilisé pour l'instant ne permet pas de choisir la solution à coût minimal parmi celles-ci. Il faudrait donc adapter le modèle afin de prendre en compte plus efficacement le coût associé au chevauchement d'activités.

Aussi, l'augmentation des coûts reliés à l'exécution du projet peut réduire le bénéfice associé. Une autre approche consisterait à maximiser le bénéfice. En effet, la réduction du temps d'exécution du projet permet d'augmenter le bénéfice en réduisant les pénalités de retard, ou en encaissant des primes. A l'inverse, cette réduction du temps implique une augmentation des coûts d'exécution. Maximiser le bénéfice reviendrait donc à trouver le meilleur compromis entre la réduction du temps et l'augmentation des coûts.

Ces deux approches montrent bien les limites actuelles de notre modèle à prendre en compte des problématiques plus globales d'exécution de projet.

3.6.2 Temps de calcul

Une autre limite liée à notre modèle d'ordonnancement est le temps de calcul nécessaire à l'établissement de l'échéancier. En effet, le temps de calcul associé à l'ordonnancement des projets avec de fortes contraintes de ressources ($RS=0,5$) est dans la plupart des cas supérieur à 5000s. Les temps de calculs reliés à quelques instances sont présentés dans le Tableau 4.3 dans la suite du mémoire.

Ces temps importants de résolution pour des projets de 30 activités limitent grandement l'applicabilité de ce modèle à des projets de plus grande envergure tels que des projets internationaux. Il faudra proposer une approche plus adaptée permettant d'obtenir un échéancier réalisable rapidement. Pour cela, il devient nécessaire de recourir à des méthodes de résolution approchée.

3.7 Conclusion

Dans ce chapitre, j'ai présenté les hypothèses et proposé un premier modèle d'ordonnancement. J'ai pu montrer l'influence des paramètres définis sur l'efficacité du chevauchement comme technique d'accélération de projet. Cependant, de nombreuses limites sont apparues. Les extensions du modèle de base permettant de pallier certaines de ces limites sont d'ailleurs présentées dans la section suivante.

CHAPITRE 4 EXTENSIONS DU MODELE DE BASE

4.1 Introduction

Dans cette section, les différentes approches développées pour pallier les limites associées au modèle de base du chapitre précédent seront abordées. Comme dans le chapitre précédent, les cas d'études utilisés seront d'abord présentés. Par la suite, les différentes approches d'optimisation et les méthodes de résolution approchées retenues seront exposées.

4.2 Cas d'études considérés

Dans ce chapitre, le plan d'expérience fut limité à neuf cas d'étude. Ainsi, trois réseaux d'activités furent générés avec les paramètres suivants:

$$\text{Paramètres de génération du réseau de projet} \quad \begin{cases} C=1,8 \\ RF=0,5 \end{cases}$$

Pour chaque réseau d'activités, on génère les paramètres de chevauchement avec la méthode exposée dans le chapitre précédent. En se basant sur l'étude précédente, en particulier l'efficacité du processus de chevauchement, on choisit des paramètres (T et C_{max}) qui permettent d'obtenir des résultats significatifs tout en limitant les temps de calculs. Les paramètres suivants ont été retenus :

$$\text{Paramètres de chevauchement} \quad \begin{cases} T=40\% \\ C_{max}=75\% \\ B=40\% \end{cases}$$

Les paramètres de génération des données de chevauchement ont permis d'obtenir des données de chevauchement de la forme exposée dans la Figure 3.2. Il faut rappeler que ces données servent uniquement à établir l'intérêt d'une approche d'accélération de projet via le chevauchement.

Une fois les trois réseaux d'activités définis, avec leurs paramètres de chevauchement et leurs demandes de ressources, on applique à chacun d'entre eux trois types de contraintes de ressources, correspondant à des contraintes fortes, moyennes et faibles. Ces contraintes sont définies par la valeur de RS associés, respectivement égale à 0,5, 0,75 et 1. On obtient alors trois instances de projet différentes pour chaque réseau d'activités. On considérera par la suite ces neuf cas d'études.

4.3 Différentes approches d'optimisation

Dans cette section, on présente les différents modèles d'optimisation permettant de prendre en compte les coûts reliés au chevauchement d'activités de manière plus efficace que le modèle de base. Les modèles proposés sont des modèles de programmation linéaire en nombres entiers, et pour chacun d'eux, les résultats associés ainsi que leurs limites seront présentés.

4.3.1 Prise en compte du coût associé au chevauchement

4.3.1.1 Modèle

Nous devons prendre en compte le coût associé au chevauchement. D'après les hypothèses posées précédemment, le coût associé au chevauchement est le coût supplémentaire d'utilisation de la main d'œuvre pour la durée des retouches. On peut donc définir le coût supplémentaire associé à la retouche de l'activité j dans le mode de précéence m comme suit :

$$C_{jm} = \sum_{t=EF_j}^{LF_j} \sum_{k \in R} X_{jtm} * R_{jk} * r_{jm} * \theta_k$$

Le coût supplémentaire total associé à l'exécution de la stratégie de chevauchement comme stratégie d'accélération de projet est donc égal à :

$$C = \sum_{j \in S} \sum_{m=1}^{m_j} C_{jm}$$

Dans la suite du document, on considérera ce coût supplémentaire total C .

Il est alors nécessaire de modifier l'objectif du problème de programmation linéaire en nombres entiers présenté dans la section précédente afin de prendre en compte le coût associé à la stratégie

de chevauchement, et ainsi de déterminer parmi les plannings qui ont une durée d'exécution minimale, celui dont le coût associé est aussi minimal :

$$\text{Minimiser } \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} + \delta * C$$

où δ est choisi de telle manière que, quelle que soit la stratégie de chevauchement et le coût associé à l'exécution du projet, $\delta * C < 1$

Cette contrainte permet de prioriser la fonction objectif : on minimise d'abord le temps d'exécution du projet, pour lequel on recherche la solution à coût minimal. Cela est possible car

$$\sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} \text{ est une valeur entière.}$$

Les mêmes contraintes que celles du modèle de base sont conservées.

4.3.1.2 Résultats

Le Tableau 4.1 montre les réductions de coûts associés à la prise en compte du coût dans l'objectif de minimisation du modèle. On observe que les coûts associés aux solutions optimales sont réduits en moyenne de 30% par rapport aux solutions trouvées dans le modèle de base. Il est donc important de considérer les coûts dans la fonction objectif du problème afin de limiter les chevauchements inutiles d'activités, et donc le rajout de retouches supplémentaires. Le problème de base présenté dans le chapitre 2 renvoyait la première solution à temps minimal, sans considérer les coûts du projet, et donc les chevauchements inutiles.

Tableau 4.1 : Diminution des coûts liés à l'optimisation des coûts

Instance	Modèle de base		Modèle de base + Optimisation des coûts		
	Temps	Coûts	Temps	Coûts	Réduction des coûts
1	104	15 300,00 \$	104	12 300,00 \$	20%
2	104	15 300,00 \$	104	13 100,00 \$	14%
3	107	8 500,00 \$	107	6 300,00 \$	26%
4	83	21 200,00 \$	83	10 800,00 \$	49%
5	85	15 000,00 \$	85	9 200,00 \$	39%
6	99	12 200,00 \$	99	10 200,00 \$	16%
7	95	20 500,00 \$	95	11 400,00 \$	44%
8	98	14 700,00 \$	98	9 900,00 \$	33%
9	107	11 000,00 \$	107	6 700,00 \$	39%

Cependant, ce modèle ne prend pas en compte la possibilité qu'une accélération du projet ne soit plus rentable à partir d'un certain point à cause des coûts importants qu'elle engendre. Dans la réalité, il est parfois plus intéressant de faire un compromis entre l'accélération de l'exécution du projet et l'augmentation du prix associé. On présentera un modèle de compromis dans la section suivante.

4.3.2 Compromis entre accélération et augmentation du coût de projet

4.3.2.1 Modèle

Afin de proposer une approche plus réaliste d'une stratégie d'accélération de projet, on propose dans cette section la recherche du meilleur compromis entre diminution du temps d'exécution du projet et augmentation du coût de réalisation. On introduit un coût d'opportunité qui modélise les primes ou pénalités liées à une avance ou un retard dans la livraison du projet comme illustré dans la Figure 4.1. Le temps de référence correspond à la date de livraison planifiée du projet.

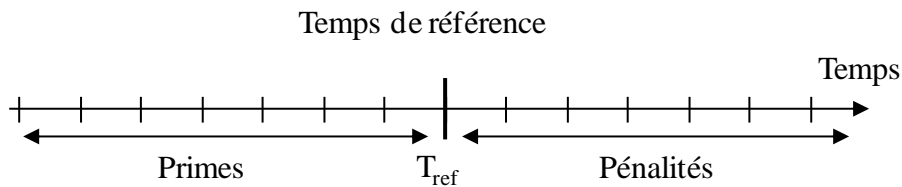


Figure 4.1 : Primes et pénalités associées à l'exécution du projet

Pour notre modèle, on fait l'hypothèse que les primes et pénalités sont proportionnelles au temps gagné, ou perdu. Dans la pratique, leurs valeurs seront établies en fonction des termes du contrat signé avec le client par exemple. Les gains, ou les pénalités, associés au temps d'exécution du projet seront égaux à :

$$G = \lambda * (T_{ref} - \sum_{t=EF_{n+1}}^{LF_{n+1}} t * X_{n+1,t,1})$$

où T_{ref} est le temps de référence obtenus lors de l'exécution du projet sans chevauchement. Dans la suite des travaux on fixe le coût d'opportunité λ à une valeur de 1000\$ par période de temps.

À ces gains, il faut retrancher le coût associé au chevauchement pour obtenir le bénéfice supplémentaire réel dégagé par l'exécution du projet en régime accéléré. Le nouvel objectif associé à ce problème de maximisation des gains du projet est donc :

$$\text{Maximiser } \lambda * (T_{ref} - \sum_{t=EF_{n+1}}^{LF_{n+1}} t * X_{n+1,t,1}) - C$$

où C est le coût défini dans la section précédente. Les contraintes associées à cet objectif restent les mêmes que celles définies dans les modèles précédents.

Nous allons voir dans la section suivante l'influence du modèle sur les solutions trouvées par le programme d'optimisation.

4.3.2.2 Résultats

Le Tableau 4.2 présente les valeurs des bénéfices obtenus suite à l'application du modèle avec optimisation des coûts, vis-à-vis de modèles de compromis Temps\Coûts. On observe que les bénéfices obtenus dans le second cas sont bien plus importants dans la plupart des cas. Ce modèle permet aussi d'assurer que les bénéfices ne seront pas négatifs, c'est-à-dire que l'accélération du projet à l'aide du chevauchement ne causera jamais de perte d'argent. L'augmentation des bénéfices se fait en contrepartie d'une augmentation des temps d'exécution des projets. On observe une augmentation des temps d'exécution de 3% en moyenne par rapport au modèle précédent. Cette augmentation de 3% du temps d'exécution permet de réduire les coûts associés aux retouches de 60% en moyenne.

Tableau 4.2 : Comparaison entre les bénéfices associés au modèle de base et ceux issus du compromis entre accélération et augmentation des coûts reliés au projet

Instances	Bénéfices associés au modèle avec optimisation des coûts	Bénéfices associés au modèle de compromis Temps\Coûts
1	4 700,00 \$	9 900,00 \$
2	3 900,00 \$	9 900,00 \$
3	7 700,00 \$	8 300,00 \$
4	4 200,00 \$	6 600,00 \$
5	3 800,00 \$	6 200,00 \$
6	-3 200,00 \$	700,00 \$
7	5 600,00 \$	7 100,00 \$
8	4 100,00 \$	5 500,00 \$
9	-1 700,00 \$	- \$

Ce modèle apparaît comme plus réaliste que le modèle précédent car il permet d’appréhender une problématique plus globale de maximisation des gains. Le problème précédent n’intégrait pas les notions de prime et de pénalité de retard qui font pourtant partie intégrante de la réalité des projets.

Cependant, les deux modèles précédents présentent de nombreuses limites, que je vais illustrer dans la section suivante.

4.3.3 Limitations

De la même manière que le modèle de base, les principales limites associées à ces modèles d’optimisation sont les temps de calculs associés. Le Tableau 4.3 expose les temps de calculs associés à la résolution des différents modèles pour les instances considérées. Les temps de calcul supérieurs à 5000s sont marqués d’une étoile. Pour ces instances, j’ai résolu le problème en plusieurs étapes pour réduire le temps de calcul. On remarque que les temps de calculs associés au modèle d’optimisation du temps d’exécution du projet avec prise en compte des coûts sont bien plus importants que ceux observés pour le modèle de base. Il faut en effet explorer l’ensemble des solutions à temps minimal pour trouver la solution à coût minimal. Dans le cas du compromis entre réduction du temps et augmentation de coûts, on observe que le temps de résolution est plus rapide que pour les autres modèles.

Cependant, dans tous les cas, ces modèles sont difficilement applicables à des projets de plus d'une centaine d'activités à cause de l'augmentation rapide du temps de calcul. Dans le but de limiter le temps de calculs de l'établissement d'un échéancier en régime accéléré pour de gros projets, une approche de résolution approchée est présentée à la section suivante.

Tableau 4.3 : Temps de calculs (en seconde) associés à la résolution des modèles présentés

		Modèle de base	Scénario 1	Scénario 2
1	0,5	60	5000*	20
	0,7	22	14	12
	1	8	7	7
2	0,5	5000*	5000*	2110
	0,7	26	86	20
	1	12	84	8
3	0,5	120	5000*	20
	0,7	13	99	13
	1	9	4	7

4.4 Approche de résolution approchée

La méthode de résolution et ses caractéristiques seront présentées dans les sections suivantes.

4.4.1 Méthode

Une approche de résolution du problème d'ordonnancement consiste à utiliser des règles heuristiques. Cette approche permet de limiter le temps de résolution mais altère la qualité de la solution. En effet, la solution trouvée n'est pas optimale comme dans les modèles précédents.

La Figure 4.2 présente l'algorithme développé, inspiré des méthodes de compression d'activités (Gray et al., 2007). La première étape consiste à ordonnancer le projet sans chevauchement d'activités afin d'obtenir un plan de référence. Ensuite, les activités sont chevauchées de manière incrémentale selon un ordre de priorité. Cet ordre de priorité est déterminant quant à la qualité de la solution finale. A chaque étape, une décision est prise en fonction de la valeur de la fonction d'évaluation qui compare le nouveau plan au plan de référence. Si l'évaluation est positive, alors le plan actif devient le nouveau plan de référence. Sinon, l'activité est supprimée de la liste des activités chevauchables. L'algorithme s'arrête lorsqu'il n'y a plus d'opportunité de

chevauchement dans la liste. Les différentes fonctions d'évaluation abordées et les règles de priorité seront présentées dans les sections suivantes.

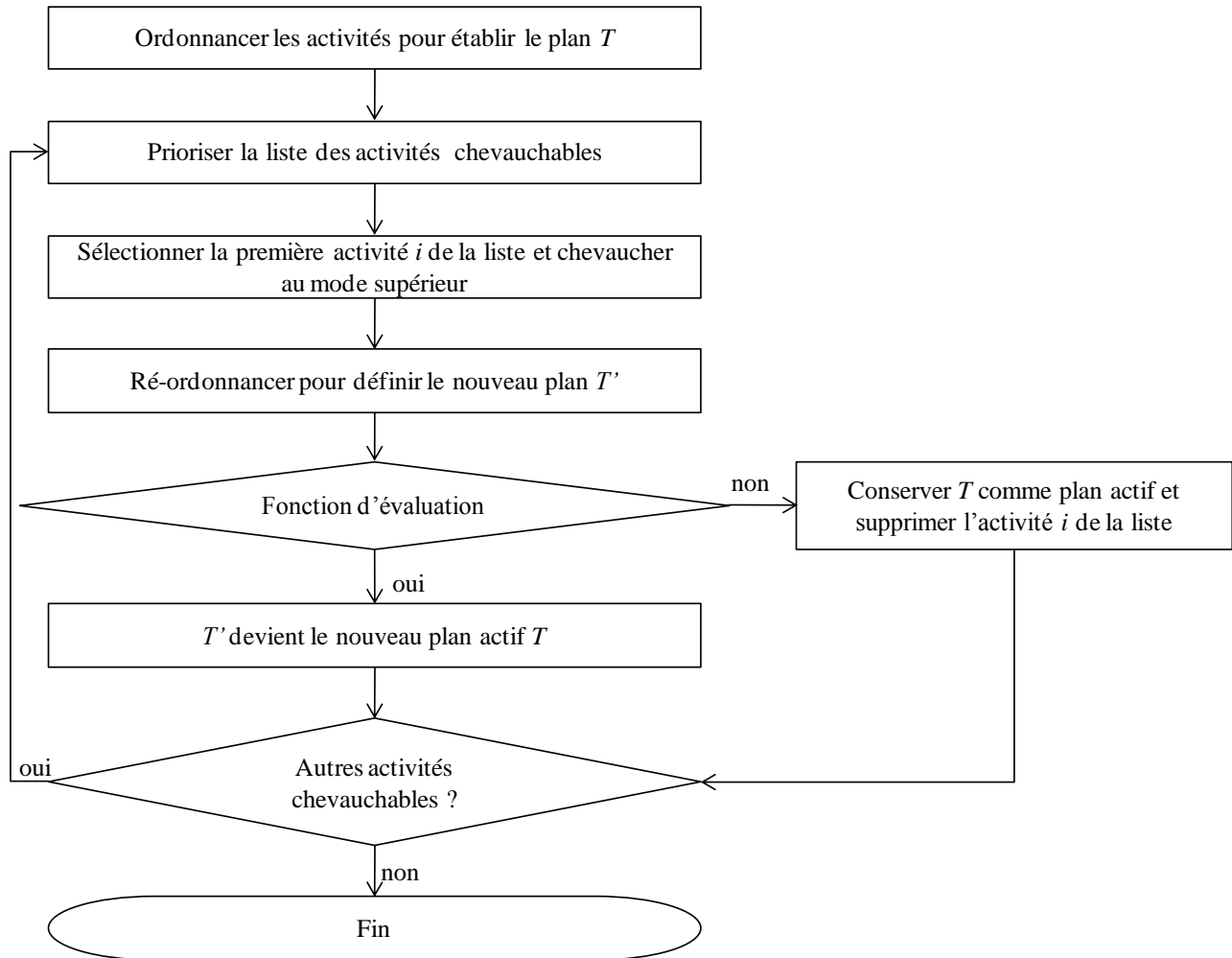


Figure 4.2 : Algorithme d'accélération de projet avec le chevauchement d'activités

4.4.1.1 Fonctions d'évaluation

Les fonctions d'évaluation sont adaptées aux scénarios considérés. Le problème général est de diminuer le temps d'exécution du projet. Nous avons vu précédemment que pour le cas d'un problème de programmation linéaire en nombres entiers, on peut extraire trois types de scénarios reflétant la réalité du projet considéré.

Le premier ne considère pas les coûts additionnels liés au chevauchement d'activités et aux retouches associées. Dans ce cas, on optimise uniquement le temps d'exécution du projet. La

deuxième approche prend en compte les coûts associés aux retouches estimées, liées au chevauchement d'activités. Dans ce cas, on cherche la solution à coût minimum parmi les solutions qui permettent l'exécution du projet en un temps minimum.

Dans le cas d'une résolution approchée, on ne peut pas différencier ces deux premiers scénarios. La résolution approchée ne permet pas d'obtenir un ensemble de solutions à temps d'exécution minimal, et donc de choisir celle dont le coût associé est le moins élevé. Dans le cadre d'une résolution approchée telle que présentée, on ne considère pas de retour en arrière possible dans les prises de décision, ni de comparaisons entre deux choix possibles. Lorsqu'on applique l'algorithme de résolution, les deux scénarios sont équivalents car le premier choix est fait sur le critère de minimisation du temps d'exécution du projet. Pour ce scénario, la fonction d'évaluation est $D(T)-D(T')$ où $D(T)$ et $D(T')$ correspondent respectivement à la durée d'exécution des plans T et T' . Le test suivant compare les deux plans à l'aide de la fonction d'évaluation :

Si $D(T) - D(T') \geq 0$, alors le plan T' devient le plan actif

Ce scénario sera considéré par la suite comme le scénario 1.

Dans le cas du scénario de compromis Temps\Coût présenté précédemment, qui correspond à une maximisation des gains associés à l'exécution du projet, on peut adapter la fonction d'évaluation de l'algorithme. On considère un coût d'opportunité qui correspond aux primes ou pénalités associées à une clôture en avance, ou en retard du projet. Dans ce cas, comme pour l'approche de résolution exacte, on ne prend pas de décision sur le temps d'exécution du projet ni sur le coût associé aux retouches. On considère uniquement le gain lié à la diminution du temps d'exécution du projet auquel on retranche les coûts associés aux retouches liées à la stratégie de chevauchement d'activités. La ligne suivante est le test utilisant la fonction d'évaluation $\lambda * [D(T) - D(T')] + C(T) - C(T')$, où $C(T)$ et $C(T')$ sont respectivement les coûts associés à l'exécution des plans T et T' .

Si $\lambda * [D(T) - D(T')] + C(T) - C(T') \geq 0$, alors le plan T' devient le plan actif

Ce scénario sera considéré par la suite comme le scénario 2.

Ces deux fonctions d'évaluation permettent d'adapter l'algorithme au contexte réel d'exécution du projet. Dans la section suivante, je vais présenter les différentes règles de priorité utilisées pour faire les choix des couples d'activités à chevaucher lors de l'exécution de l'algorithme.

4.4.1.2 Règles de priorité

Dans ce modèle, les activités chevauchables sont priorisées selon des règles de priorités. Dans le cas présent, deux règles de priorité furent retenues à titre de démonstration.

Tout d'abord, les activités sont classées en fonction de leur marge totale ; celles qui ont les marges totales les plus petites sont placées en premier. La marge totale est la différence des dates de fin d'une activité entre la planification au plus tard et la planification au plus tôt des activités; ces planifications étant calculées sans prendre en compte les contraintes de ressources. Les activités dont la marge totale est nulle sont les activités critiques du projet, elles ont une influence directe sur le temps d'exécution du projet. On ne peut cependant pas identifier de chemin critique du fait de la prise en compte des contraintes de ressources. En cas d'égalité de la marge totale entre plusieurs activités, on priorise les activités dont la quantité de ressources nécessaire à leur réalisation est la plus faible. Les activités mobilisant le moins de ressources sont en effet celles qui auront le moins d'impact sur les contraintes de ressources. Enfin, en cas d'égalité, on classe les activités en fonction de leur nombre de successeurs, plus celui-ci est important, plus les activités seront mises en avant. On considérera par la suite cette règle comme la première règle de priorité. La liste de priorité est actualisée est mise à jour à chaque itération.

L'autre règle de priorité considérée consiste à intervertir les deux dernières règles de tri de la règle précédente. On triera alors selon la marge totale croissante, puis le nombre de successeurs croissant et enfin la quantité de ressources requise décroissante pour l'exécution normale de l'activité. On considérera cette règle comme la seconde règle de priorité.

On compare les résultats associés à ces deux règles de priorité dans les sections suivantes.

4.4.1.3 Implantation

L'algorithme fut implanté dans un progiciel de gestion de projet, ici MS Project 2007. MS Project est un progiciel de gestion de projet très répandu dans l'industrie. Les performances de son outil d'ordonnancement sont toutefois limitées. En effet, des études statistiques ont montré qu'en moyenne, on obtient un écart de 5,35% par rapport au temps d'exécution optimal, avec un écart type de 6% (Kolisch 1999). La qualité des résultats est donc non prévisible. Ceci permet toutefois de tester l'approche proposée sur un outil largement utilisé dans la pratique.

L'algorithme d'accélération de projet a été développé sous la forme d'une macro Visual Basic implantée dans Excel avec une liaison avec MS Project. Le programme est présenté en annexe 3.

Les sections suivantes font l'analyse des résultats obtenus.

4.4.2 Analyse des résultats

Dans cette section, on présente les résultats obtenus à l'aide de l'heuristique présentée. Dans un premier temps, l'influence des règles de priorité sera analysée afin de choisir la meilleure configuration pour l'analyse des résultats subséquente.

4.4.2.1 Influence des règles de priorité

Les résultats obtenus avec les deux types de règles de priorité sont similaires. Lorsque l'on compare la seconde règle de priorité avec la première, seule une instance donne un résultat supérieur au résultat précédent, les autres étant égales. Cependant, cette seconde configuration a une influence sur le temps de résolution, augmentant en général le nombre d'itérations nécessaires à l'obtention de la solution. Ce constat démontre l'importance du choix de la règle de priorité en fonction de la nature des projets étudiés.

4.4.2.2 Résultats

Les temps de référence, correspondant aux temps d'exécution du projet sans chevauchement d'activités, donnés par MS Project ne sont pas les temps d'exécution optimaux. Pour chaque projet, le gain par rapport au temps de référence donné par MS Project et non par rapport à l'optimal sera évalué.

Dans un premier temps, les résultats associés au scénario 1, c'est-à-dire à l'utilisation de la fonction d'évaluation visant à minimiser le temps d'exécution du projet sans considérer l'augmentation des coûts, est présenté.

On observe à la Figure 4.3 que les gains obtenus sur le temps d'exécution du projet sont moins importants en moyenne (10% au lieu de 12%) et plus aléatoires que les gains obtenus par la méthode d'optimisation alors que les coûts sont comparables. Le nombre d'itérations nécessaires à l'obtention des résultats est inférieur à 150 pour l'ensemble des instances considérées ; chaque itération impliquant le calcul d'un ordonnancement du projet.

Cette première constatation permet de confirmer l'efficacité de l'algorithme pour accélérer l'exécution du projet et cela en un temps de calcul raisonnable.

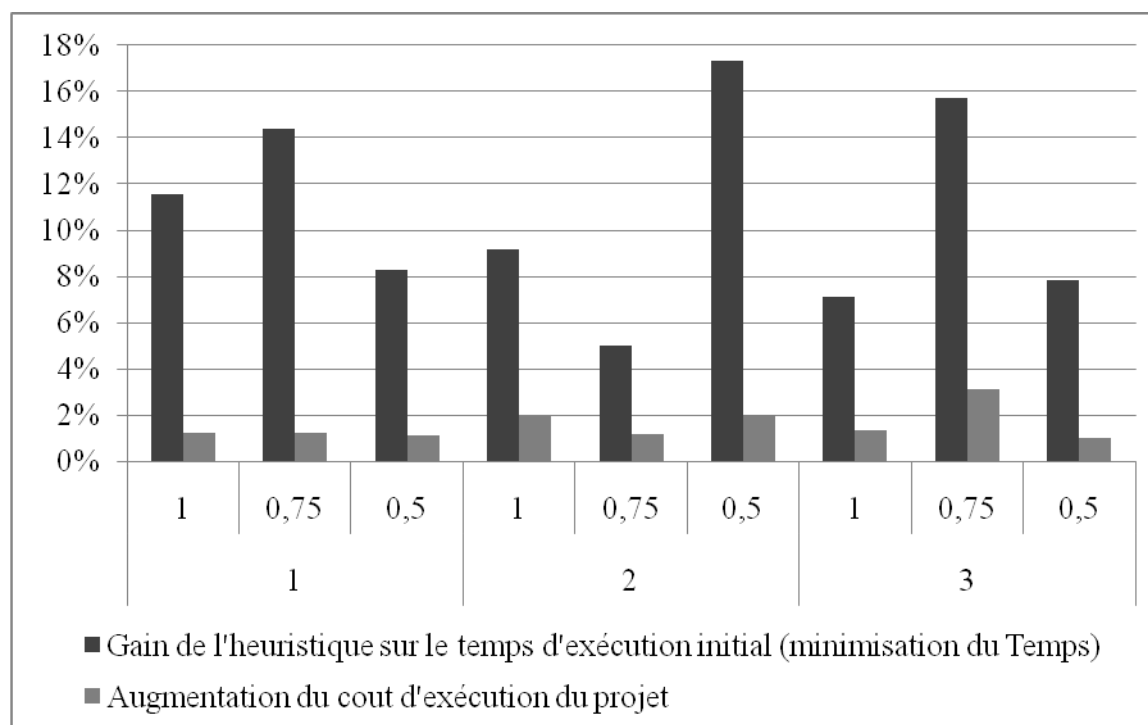


Figure 4.3 : Gain et coût associés à l'exécution de l'algorithme avec la fonction d'évaluation du scénario 1

Cependant, la Figure 4.3 ne permet pas d'évaluer la qualité des solutions obtenues. En effet, pour certaines instances, on remarque que le gain sur le temps d'exécution peut être plus important avec l'heuristique, par exemple pour l'instance 2 avec $RS=0,5$, le gain observé est de 17% contre 7% à l'optimal. Cela s'explique par la valeur du temps de référence donnée par MS Project. Les gains observés précédemment ne sont pas comparables aux gains de la première méthode car ils ne se basent pas sur le même temps de référence.

A la Figure 4.4, on observe la qualité des solutions en les comparant aux solutions optimales. Une valeur nulle signifie que la valeur trouvée par l'heuristique est égale à la valeur optimale. Une valeur non nulle illustre la marge d'erreur entre la valeur optimale et la valeur obtenue à la suite de l'exécution de l'algorithme.

La Figure 4.4 fait donc apparaître les écarts entre les temps de référence et les solutions obtenues par l'algorithme proposé. On observe tout d'abord que, pour les temps de référence, l'erreur est

d'autant plus importante que les contraintes de ressources sont sévères. Cela s'explique par la faible performance de l'outil d'ordonnancement utilisé (MS Project). La moyenne d'erreur pour les temps de référence est de 5% par rapport à l'optimal. Cette valeur est comparable avec les résultats donnés dans des études précédentes (Kolisch, 1999). La qualité observée des solutions suite à l'application de l'algorithme est variable, et la marge d'erreur moyenne par rapport à l'optimal est de 6%. En comparaison avec les temps de référence, il n'y a donc qu'un pourcent d'erreur supplémentaire induit par l'utilisation de l'algorithme pour l'obtention d'un échancier accéléré prenant en compte les modes de chevauchement et les temps de retouche associés.

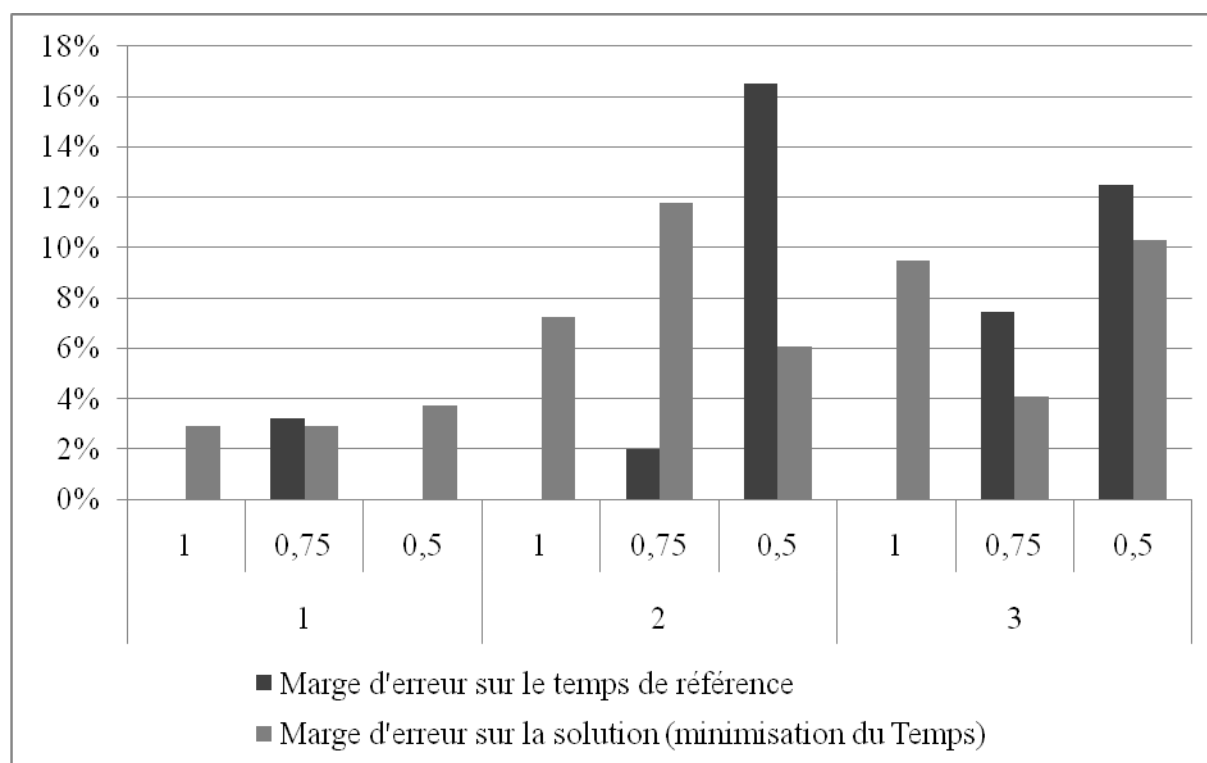


Figure 4.4 : Marges d'erreur associées à l'heuristique avec la fonction d'évaluation du scénario 1

On remarque aussi que l'utilisation de l'algorithme permet dans certains cas de compenser la faible performance de l'outil d'ordonnancement de projet, comme par exemple pour l'instance 2 avec $RS=0,5$. La marge d'erreur observée passe de 16% pour le temps de référence à 6% pour la solution. Aussi, dans nos résultats, l'utilisation de l'algorithme induit une erreur comprise entre 3% et 10% par rapport à l'optimal.

Le Tableau 4.4 illustre le nombre d'itérations nécessaires pour atteindre la meilleure solution à l'aide de l'algorithme. Chaque itération correspond à un ordonnancement de projet. On observe

que le nombre d'itérations est assez faible et aléatoire. Aussi, l'utilisation d'une heuristique permet de s'affranchir de la complexité du problème RCPSP, le temps de résolution ne sera pas directement lié avec la taille de problème. L'application de l'heuristique permet donc d'obtenir une solution de bonne qualité dans un temps raisonnable.

Tableau 4.4 : Nombre d'itérations nécessaires à l'obtention de la solution

Réseaux		1	2	3
RS	1	13	67	94
	0,75	14	5	135
	0,5	17	22	3

Une analyse semblable peut être faite pour la méthode heuristique avec une autre fonction d'évaluation. On utilisera ici la fonction d'évaluation présentée précédemment permettant de faire un compromis entre la réduction du temps d'exécution du projet grâce au chevauchement et l'augmentation des coûts. Cette fonction correspond au scénario 2.

On observe ici que cette fonction d'évaluation permet de se rapprocher plus efficacement de la solution optimale. La marge d'erreur sur la solution est de 4% en moyenne, contre 6% précédemment (voir Figure 4.5 et Figure 4.6). On remarque aussi que la marge d'erreur de 4% est inférieure à la marge d'erreur initiale observée dans MS Project lors de l'ordonnancement de projet avec contraintes de ressources. Le processus mis en place permet d'améliorer les performances du logiciel d'ordonnancement de projet.

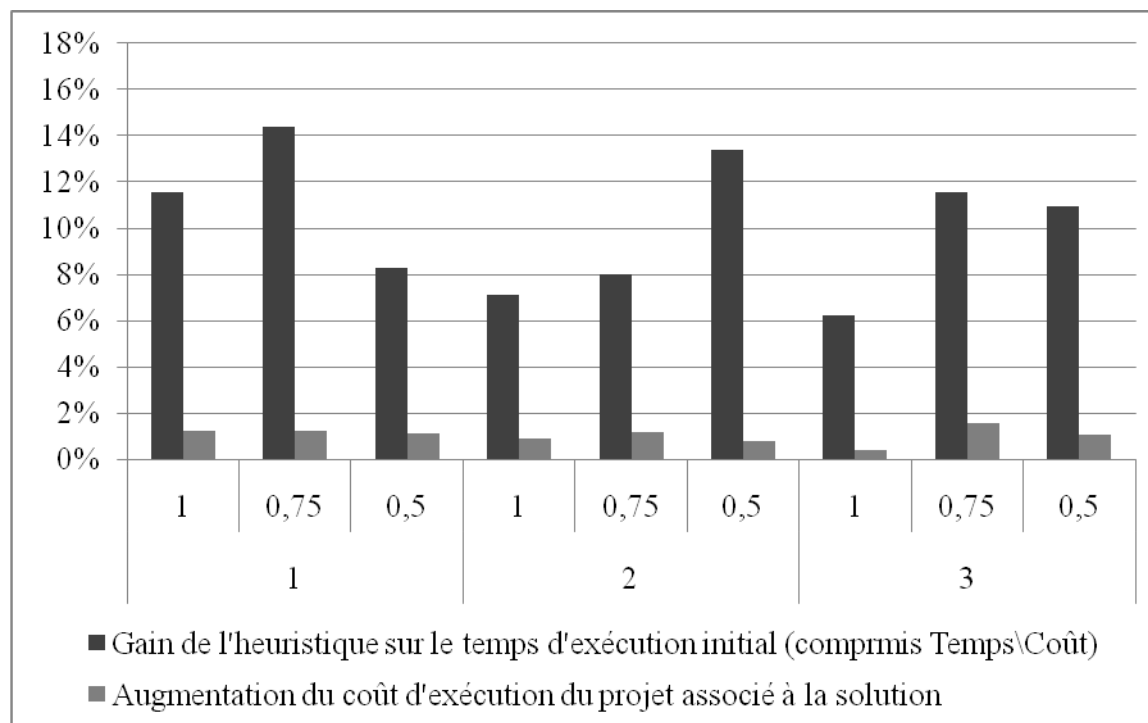


Figure 4.5 : Gain et coût associés à l'exécution de l'algorithme avec la fonction d'évaluation du scénario 2

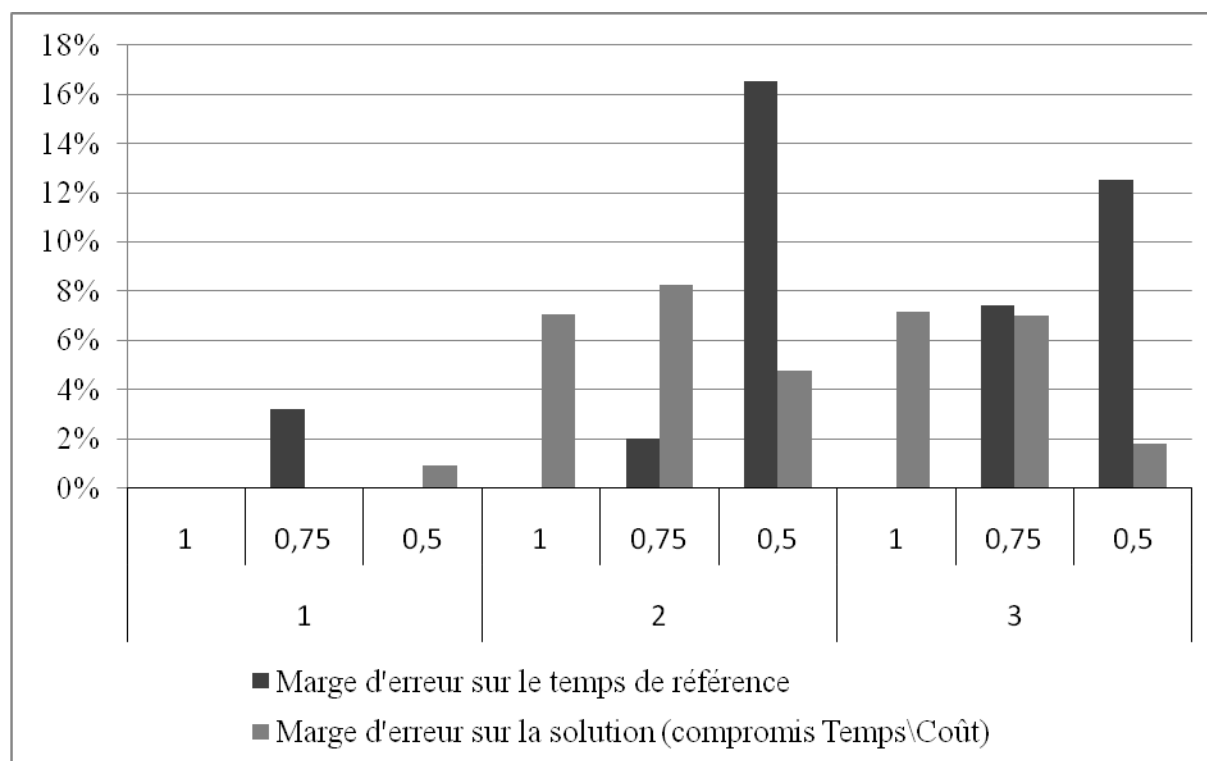


Figure 4.6 : Marges d'erreur associées à l'heuristique avec la fonction d'évaluation du scénario 2

4.4.3 Limitations

La résolution approchée à l'aide d'un algorithme heuristique présente cependant des limites. Tout d'abord, on observe que la qualité des solutions est très variable avec un écart type de 3% pour chacun des scénarios envisagés.

On remarque aussi que la nature de l'algorithme rend les retours en arrière impossible et donc ne permet pas de revenir sur une prise de décision qui peut s'avérer par la suite inefficace, où trop coûteuse. La simplicité de l'algorithme peut ainsi être remise en cause.

On note aussi que l'algorithme a été développé à l'aide d'une macro Visual Basic dans Excel puis relié à MS Project. Une implantation directe dans MS Project permettrait d'obtenir de meilleures performances.

Enfin, cet algorithme d'accélération dépend en grande partie des performances du progiciel de planification qui est utilisé. Il pourrait être pertinent de développer une méthode heuristique d'ordonnancement performante pour s'affranchir de la qualité variable des logiciels utilisés.

4.5 Conclusion

Dans ce chapitre, nous avons proposé différentes extensions permettant de pallier les limitations du modèle de base. Les modifications apportées sur le modèle de base permettent d'appréhender avec plus de fidélité les problématiques rencontrées dans la réalité des projets d'ingénierie. La prise en compte des coûts associés au chevauchement d'activités est essentielle lors de l'optimisation du planning. Les résultats ont montré que la considération des coûts permet de réduire de façon importante les retouches, et d'obtenir des solutions dont la facture est réduite de 30% en moyenne. Ce modèle d'ordonnancement est valable dans un contexte de projet où l'on cherche avant tout à diminuer le temps d'exécution.

Dans un cas plus modéré, où l'objectif principal est de maximiser les gains obtenus suite à l'exécution du projet, le modèle de compromis est plus adapté. Nous avons vu que l'accélération de projet à l'aide du chevauchement peut causer de nombreuses retouches, et donc une augmentation, parfois importante du coût d'exécution du projet. Cette augmentation est parfois telle qu'elle peut finalement pénaliser le projet, et les bénéfices escomptés. De ce point de vue, il est important de rechercher le meilleur compromis entre la réduction du temps d'exécution et l'augmentation des coûts associés au chevauchement d'activités. Les résultats associés au

modèle proposé ont montré qu'il permet d'augmenter grandement les bénéfices associés au projet. Aussi, un tel modèle assure que l'accélération du projet n'engendrera pas de perte.

Ces deux modèles sont cependant limités par un temps de calcul trop important. Ce temps de calcul rend l'application impossible pour des projets de plus grande ampleur, pour les mêmes raisons que le modèle de base proposé dans le chapitre précédent. Cependant, ils permettent d'appréhender plus fidèlement l'efficacité du chevauchement comme technique d'accélération de projet.

Aussi, une approche de résolution approchée, qui permet de résoudre le problème d'ordonnancement, fut développée. Cette méthode permet d'approcher avec une précision acceptable les solutions optimales. De plus, elle utilise des logiciels de gestion de projet communément utilisés dans la pratique ce qui rend son application plus facilement envisageable. Malgré tout, une telle approche peut parfois être irréaliste car elle demande d'avoir à disposition de nombreuses données d'entrée. Le prochain chapitre discutera d'ailleurs de la pertinence pratique des modèles développés dans cette étude en plus de traiter des limitations et différentes opportunités futures de recherche.

CHAPITRE 5 DISCUSSION

5.1 Introduction

Une méthode générale d'accélération de projet à l'aide du chevauchement d'activités, les différents modèles d'ordonnancement et les résultats obtenus ont été présentés dans les chapitres précédents. On peut remarquer que malgré nos efforts de modélisation, les modèles proposés présentent toujours certaines limites. Ce chapitre va étendre la discussion plus globalement et aborder la pertinence de l'étude, ses limitations ainsi que les opportunités futures de recherche.

5.2 Pertinence de l'étude

On a pu voir dans le premier chapitre tout particulièrement, que malgré les nombreuses études effectuées dans le domaine de l'ordonnancement, et l'accélération de projet à l'aide du chevauchement d'activités, peu d'entre elles proposaient une approche applicable dans un contexte d'entreprise. En ce sens, l'étude proposée est réaliste, car elle propose un processus complet d'accélération de projet avec chevauchement d'activités, considérant à la fois l'acquisition des données de chevauchement et l'ordonnancement du projet.

Tout d'abord, dans le cadre de cette recherche, nous avons pu rencontrer des professionnels travaillant chez SNC-Lavalin dans les départements de planification et d'ingénierie. Ces rencontres ont permis de confirmer l'intérêt d'une étude sur les impacts du chevauchement sur l'exécution d'un projet. Dans la pratique, les professionnels utilisent en permanence le chevauchement d'activités pour accélérer l'exécution des projets. Les relations de précédence classiques fin-début utilisées dans la littérature sont très peu utilisées dans des projets d'ingénierie. Les professionnels ne disposent cependant d'aucun outil d'aide à la décision leur permettant d'évaluer la pertinence ou non des chevauchements d'activités. Il n'y a donc pas d'outil qui permet de mesurer l'impact du chevauchement. Les planificateurs se réfèrent avant tout à leurs expériences et aux historiques disponibles de projets similaires. Le développement d'une procédure permettant d'aider la prise de décision vis-à-vis des chevauchements d'activités est donc pertinent pour les professionnels pour aider à la bonne planification des projets en régime accéléré. Le développement d'une telle procédure passe d'abord par la compréhension des mécanismes de chevauchement et de leurs impacts globaux. Pour cela, l'approche ici proposée a

été de développer et de tester un modèle de base d'ordonnancement en programmation linéaire en nombres entiers avec chevauchement d'activités sur de nombreuses instances générées aléatoirement afin d'identifier les paramètres clé pour l'efficacité du chevauchement comme technique d'accélération. Dans un second temps, une extension du modèle de base et une approche heuristique ont été proposées afin se rapprocher des problématiques rencontrées dans la réalité et de limiter le temps de calculs.

Les études menées sur les modèles de programmation linéaire en nombres entiers permettent d'obtenir les solutions optimales des différents problèmes. L'application de ces modèles sur un grand nombre de projets comparables a permis d'identifier les caractéristiques de projets propices à l'accélération de projet via le chevauchement d'activités. Ces enseignements, même théoriques, et limités à des projets de petite taille, donnent des informations précieuses sur les impacts de chevauchement sur l'exécution d'un projet et sur l'importance à accorder à l'estimation de certains facteurs.

Le modèle de base a ensuite été adapté pour intégrer les problématiques liées aux coûts associés au chevauchement. Une méthode heuristique d'ordonnancement permettant d'obtenir des résultats approchés en un temps raisonnable a été proposée sur la base de ces extensions. Cette heuristique est facilement intégrable à un progiciel de gestion de projet, notamment MS Project, sur lequel l'algorithme a été testé. La comparaison des résultats avec les méthodes de résolution linéaires en nombres entiers et les faibles temps de calculs associés rendent légitime notre approche heuristique. Son intégration facilement envisageable à des progiciels professionnels ne donne que plus de valeur à notre approche.

Néanmoins, nos conclusions doivent être pondérées. En effet, les modèles développés et les résultats associés ne reposent que sur des données théoriques. Les échanges avec des planificateurs de projet travaillant au sein de notre partenaire industriel SNC-Lavalin ont permis de soulever des différences majeures entre la réalité et notre modélisation. Ces différences sont discutées à la section suivante.

5.3 Limitations

Les limitations de l'approche proposée sont nombreuses. Elles sont issues tout d'abord des hypothèses considérées pour développer notre approche, mais aussi des différentes méthodes d'ordonnancement présentées.

5.3.1 Hypothèses

Tout d'abord, il faut relever que certaines hypothèses peuvent paraître réductrices. La première d'entre elle, est l'hypothèse sur les flux d'informations. On considère en effet que la décomposition des activités et la réalisation de la séquence d'activités à l'aide de l'étude des DSM permet d'obtenir un flux d'information unidirectionnel. Cette hypothèse permet de ne pas considérer les retours d'informations entre les activités, et donc les itérations qu'il pourrait y avoir entre deux activités. Les relations entre activités sont, dans notre modèle, soit dépendantes, soit indépendantes et on ne considère pas d'activités interdépendantes. Cette hypothèse permet de s'affranchir des retours d'information, et donc de considérer que le chevauchement entre deux activités n'a un impact que sur l'activité en aval. Dans la réalité, les échanges d'information sont beaucoup plus nombreux, et de nombreuses activités d'ingénierie sont interdépendantes. La prise en compte de ces flux supplémentaires d'information permettrait d'obtenir un modèle plus exploitable pour des projets réels, qui contiennent des activités interdépendantes.

Dans ses travaux, on considère aussi que les données de chevauchement sont connues. Cette hypothèse, utile pour l'élaboration d'un premier modèle et pour l'étude de l'impact des chevauchements est réductrice dans le sens que l'évaluation de ces données est parfois difficile. Les rencontres avec les praticiens de SNC-Lavalin ont permis d'évaluer la difficulté d'estimer les différents paramètres de chevauchement pour l'application du modèle sur des projets réels de grande taille. Il serait possible de déterminer les différents modes de chevauchement possibles à partir des jalons identifiés lors de la définition des activités, mais il serait très difficile d'évaluer à partir de données historiques les temps de retouches associés car les retards qui peuvent être observés sur des activités chevauchées ne sont pas forcément dus au chevauchement lui-même. Ils peuvent être issus d'un changement de spécifications du client, ou de retard de livraison par exemple. Le détail et les causes de ces retards ne sont pas toujours détaillés dans les bases de données historiques. Pour ces raisons, les données historiques paraissent difficiles à exploiter.

Une évaluation plus pragmatique des données de chevauchement qui nécessiterait l'estimation des données pour chaque couple chevauchables nécessiterait une analyse beaucoup trop poussée des activités, et donc un temps beaucoup trop long pour la planification d'un projet de grande taille. En ce sens, dans le cadre de projet de grande ampleur, il apparaît impossible de déterminer l'ensemble des données nécessaires à l'application d'un tel processus d'accélération.

Enfin, la prise en compte des contraintes de ressources apparaît importante. Cependant, dans les travaux présentés, seules les contraintes de capacité des ressources ont été étudiées. Dans la réalité, il est aussi important de niveler l'utilisation des ressources afin de produire un échéancier applicable dans un contexte industriel. Le nivellement des ressources consiste à limiter les variations d'utilisation d'un type de ressources lors de la réalisation du projet. Ignorer le nivellement ne permet pas à notre modèle de prendre en compte de façon réaliste les contraintes liées à l'allocation des ressources.

Dans ce modèle, on considère aussi que l'ordonnancement est fait pour l'ensemble du projet. Les échanges avec les praticiens ont permis d'appréhender plus clairement le réel processus de planification utilisé. Dans un premier temps, un planning sommaire est établi à partir des activités principales et des informations préliminaires. Ensuite, chaque phase est décomposée en lots plus petits, pour obtenir au final des activités. Cette approche du haut vers le bas, ou descendante, implique la création de contraintes temporelles dès la création des premier échéanciers du projet quant à la réalisation de certaines activités. Notre modèle d'ordonnancement, qui se place au niveau des activités, c'est-à-dire au niveau d'ordonnancement le plus bas ne prend en compte l'existence de contraintes temporelles existantes pour certaines activités.

Finalement, dans ces travaux, toutes les données utilisées ont été considérées comme des paramètres déterministes. Une approche stochastique permettrait, à défaut de générer un échéancier applicable, de prendre en compte les variations prévisibles entre les estimations et la réalité, et donc de fournir un outil supplémentaire d'aide à la décision pour les praticiens.

On peut voir que les hypothèses prises en compte au début de cette recherche ont eu un impact direct sur le réalisme du processus proposé. Le choix des méthodes d'ordonnancement ont toutefois aussi limité la portée du modèle proposé.

5.3.2 Méthodes d'ordonnancement

Les méthodes d'ordonnancement utilisées dans mes travaux présentent de nombreuses limites déjà évoquées. La résolution de modèle de programmation linéaire en nombres entiers limite l'application du modèle d'ordonnancement à des projets de petite taille et peut nécessiter dans certains cas un temps de calcul important. On justifie cependant cette approche par la nécessité d'utiliser une base de comparaison pour les méthodes approchées développées par la suite. Quant à la méthode heuristique présentée, elle est en grande partie dépendante de la performance du logiciel de planification utilisé. Par ailleurs, l'heuristique présentée ne permet pas d'obtenir des résultats de très grande qualité, car elle n'autorise pas de retour en arrière dans les choix qui ont été faits. Par exemple, elle ne permet pas d'annuler un choix sur le chevauchement de deux activités, s'il s'avère inutile par la suite.

Les objectifs pris en compte lors de l'ordonnancement sont aussi perfectibles. Dans ce mémoire, deux types d'approches ont été présentées : l'une permettant de minimiser le temps d'exécution du projet tout en trouvant une solution à coût minimal, l'autre visant à rechercher le meilleur compromis entre la réduction du temps d'exécution du projet et l'augmentation du coût supplémentaire associé au chevauchement d'activités. Comme nous l'avons vu précédemment, aucun des modèles développés ne prend en compte la nécessité de niveler l'utilisation des ressources. Le recours à des périodes de retouche déterministes est aussi une limitation importante. Dans la pratique, la considération de la nature stochastique des activités est aussi souhaitable.

5.4 Améliorations envisageables

Des limitations importantes ont été décelées dans la section précédente. Elles sont dues en partie à des hypothèses trop réductrices. La prise en compte des retours d'information et des relations plus complexes entre les activités permettrait de proposer un modèle plus proche de la réalité. L'hypothèse d'un flux d'information unidirectionnel au sein d'un projet est difficile à rencontrer dans la réalité. Il sera donc important dans des études futures de considérer les interactions possibles entre les activités.

Aussi d'autres rencontres avec les professionnels ont fait ressortir une attente importante quant à la réalisation d'un outil permettant de relier directement la planification des activités avec les flux

d'information, notamment au sein des activités d'ingénierie. L'étude des flux d'information permettrait en effet de déterminer plus précisément les relations entre les activités et les opportunités de chevauchement ainsi que leurs impacts sur le projet. Cette opportunité de recherche permettrait de faciliter l'estimation des paramètres utiles au chevauchement d'activités.

On a pu observer la quantité importante des données nécessaires à l'application du processus d'accélération de projet. Dans cette optique, il serait intéressant de développer une approche plus globale qui consisterait à envisager les chevauchements non pas au niveau des activités, mais plutôt au niveau des disciplines d'ingénierie (Structure, Procédés, Béton, ...) et/ou des phases d'exécution du projet (Ingénierie préliminaire, Ingénierie détaillée, Exécution,...). Une étude sur des données historiques permettrait de fournir aux professionnels un outil d'aide à la décision facile à utiliser.

Enfin, intégrer le nivellement des ressources dans les objectifs de l'ordonnancement est capital pour l'élaboration d'échéanciers réalistes.

Au niveau des modèles d'ordonnancement, intégrer la méthode heuristique utilisée à d'autres logiciels de gestion de projet tels que Primavera permettrait d'avoir une vision plus globale sur l'applicabilité de la méthode proposée dans cette étude.

L'algorithme de résolution heuristique présenté a été implanté dans Excel à travers une macro en Visual Basic. Il serait intéressant d'intégrer directement cette macro dans le fonctionnement des logiciels de planification de projet comme MS Project. Cela permettrait d'améliorer la rapidité d'exécution de l'algorithme et de rendre son application plus fonctionnelle.

Aussi, il serait pertinent de proposer des approches d'ordonnancements métaheuristiques telles que des algorithmes de recherche tabou ou des algorithmes génétiques. Ces méthodes de résolution plus évoluées permettent de dépasser les optimums locaux et ainsi de trouver des solutions de meilleures qualités, tout en gardant un temps de résolution limité.

5.5 Conclusion

Ce chapitre a permis de justifier la pertinence de l'étude qui a été menée et aussi mettre en perspective les limites associées ainsi que les différentes opportunités d'amélioration identifiées. L'étude se révèle pertinente mais les hypothèses posées ont amené plusieurs limitations.

CONCLUSION

Cette étude a porté sur le développement d'un processus d'accélération de projet à l'aide du chevauchement d'activités. Son but était de répondre à un besoin des gestionnaires de projet de disposer d'outils de planification et d'aide à la décision performants et proches de la réalité. En effet, malgré l'abondance des modèles développés dans la littérature scientifique, aucun d'entre eux n'aborde d'une façon opérationnelle le chevauchement d'activités. Ces lacunes obligent les gestionnaires à baser leurs décisions relatives au chevauchement d'activités sur leurs expériences ou sur des données historiques de projets similaires lorsqu'elles sont disponibles.

Pour répondre à ce besoin, un processus général d'accélération de projet à l'aide du chevauchement d'activités a été proposé dans un premier temps. Ce processus aborde la décomposition du projet en activités, la réalisation de la séquence d'activités, l'acquisition des données nécessaires à l'ordonnancement, et l'ordonnancement en tant que tel. Dans notre approche, les différentes opportunités de chevauchement comme des modes de précédences possibles ont été considérées. Un couple d'activités peut avoir différents modes de précédence, mais lors de la planification, on en choisit un seul. A chaque mode de précédence, aussi appelé mode de chevauchement, on associe un degré de chevauchement et un temps de retouche associé, nécessaire pour s'adapter aux évolutions de l'information disponible.

Une méthode de génération de projet avec mode de chevauchement a aussi été développée dans cette étude à partir de trois paramètres. Cette méthode de génération de projet a été utile pour tester les différents modèles d'ordonnancement envisagés.

Au niveau de la résolution, un modèle de base d'ordonnancement de projet prenant en compte les opportunités de chevauchements a été développé. L'implémentation de ce modèle sur de nombreuses instances de projet a permis de vérifier l'influence des paramètres de génération de projet sur l'efficacité du chevauchement comme technique d'accélération de projet. Cette étude a permis de mettre en avant les paramètres qui ont le plus d'influence sur l'efficacité du chevauchement comme technique d'accélération de projet. Aussi, la résolution optimale du problème d'ordonnancement donne une base de référence pour comparer la qualité des solutions proposées par les autres modèles d'ordonnancement.

Des modèles plus proches de la réalité ont ensuite été envisagés. Le modèle de base ne considèrerait pas le coût associé au chevauchement. Il a été introduit dans l'objectif des deux autres modèles proposés. L'un minimise le temps d'exécution du projet, tout en recherchant la solution à coût minimal. L'autre recherche le meilleur compromis entre la réduction du temps d'exécution du projet et l'augmentation du coût associé au chevauchement d'activités. Ces deux modèles d'ordonnancement ont été résolus par un solveur de programmation linéaire en nombres entiers. La qualité des solutions a été comparée avec le modèle de base.

Ces approches de programmation linéaire comportent de nombreuses limitations, notamment en termes de temps de calcul et d'implantation dans un milieu industriel. Dans cette optique, une approche heuristique, basée sur l'utilisation d'un progiciel de gestion de projet utilisé dans la pratique, a été développée. Pour chacune des approches précédentes, minimisation du temps et compromis, nous avons adapté la méthode heuristique, puis comparé la qualité des solutions obtenues.

Les résultats obtenus sont intéressants, mais des ajustements sont encore nécessaires pour obtenir des solutions directement applicables en milieu industriel. En effet, les limitations sont multiples, de part les hypothèses qui ont été posées au début de cette étude. Cependant de nombreuses perspectives d'avenir ont été identifiées.

Notre recherche possède une portée intéressante et présente deux intérêts majeurs. Tout d'abord, elle propose une méthodologie qui permet l'acquisition des données de chevauchement de manière plus réaliste que dans la plupart des papiers qui considèrent le chevauchement avec une relation linéaire. Ensuite, les différents modèles d'ordonnancement offrent une base de comparaison à l'approche heuristique développée, qui permet une résolution en un temps raisonnable et une intégration facile au progiciel de gestion de projet. Enfin, la documentation d'une méthodologie de génération de projets avec modes de chevauchement permettra dans l'avenir de tester les modèles sur de nombreuses instances.

La contribution à la recherche de cette étude est non négligeable. Dans un premier temps, un modèle de base d'ordonnancement ainsi qu'une méthode de génération de projet ont été développés. Les résultats du modèle de base ont été présentés dans un article (en annexe 1). Puis une méthodologie globale et une approche heuristique ont été présentées lors d'une conférence à

Saint Etienne (ROADEF 2011) et un article à été soumis pour la conférence du CIGI (en annexe 2).

En adaptant un modèle simple, mais néanmoins plus réaliste que ceux présents dans la littérature et en choisissant une approche de résolution adaptée aux logiciels utilisés dans la réalité, cette étude pourra servir de base à des travaux de recherche futurs sur le sujet.

BIBLIOGRAPHIE

- Bhatt, V. (1988). Optimal Synthesis of Large Systems Usingg Decomposition Techniques and Parallel Processing The 1988 Design Automation Conference. Kissimmee, FL.
- Blazewicz, J., J. K. Lenstra, et al. (1983). "Scheduling subject to resource constraints: classification and complexity." Discrete Applied Mathematics 5(1): 11-24.
- Bogus, S. M., K. R. Molenaar, et al. (2005). "Concurrent engineering approach to reducing design delivery time." Journal of Construction Engineering and Management-Asce 131(11): 1179-1185.
- Browning, T. R. (2001). "Applying the design structure matrix to system decomposition and integration problems: A review and new directions." Ieee Transactions on Engineering Management 48(3): 292-306.
- Brucker, P., A. Drexl, et al. (1999). "Resource-constrained project scheduling: Notation, classification, models, and methods." European Journal of Operational Research 112(1): 3-41.
- Chen, S. J. and L. Li (2003). "Decomposition of interdependent task group for concurrent engineering." Computers & Industrial Engineering 44(3): 435-459.
- Cho, S. H. and S. D. Eppinger (2005). "A simulation-based process model for managing complex design projects." Ieee Transactions on Engineering Management 52(3): 316-328.
- Demeulemeester, E. L. and W. S. Herroelen (1997). "New benchmark results for the resource - Constrained project scheduling problem." Management Science 43(11): 1485-1492.
- Eppinger, S. D., D. E. Whitney, et al. (1994). "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design-Theory Applications and Concurrent Engineering 6(1): 1-13.
- Fayez, M., P. Axelsson, et al. (2003). "DSM Versus CPM: Issues for Planning Design & Construction Activities." Construction Research.
- Ger, J. E. V. and R. Y. Qassim (2008). "Project Acceleration via Activity Crashing, Overlapping, and Substitution." Ieee Transactions on Engineering Management 55(4): 590-601.

C. F. Gray, Y. Langevin, et E. W. Larson, *Management de projet*, Montréal: Chenelière McGraw-Hill, 2007.

Hartmann, S. (1999). Project scheduling under limited resources.

Hartmann, S. and D. Briskorn (2010). "A survey of variants and extensions of the resource-constrained project scheduling problem." European Journal of Operational Research 207(1): 1-14.

Hegazy, T. (2002). Computer-based construction project management. Upper Saddle River, N.J., Prentice Hall.

Hendrickson, C. and T. Au (1989). Project management for construction : fundamental concepts for owners, engineers, architects, and builders. Englewood Cliffs, N.J., Prentice Hall.

Herroelen, W. (2005). "Project scheduling - Theory and practice." Production and Operations Management 14(4): 413-432.

Jung, Y. and S. Woo (2004). "Flexible work breakdown structure for integrated cost and schedule control." Journal of Construction Engineering and Management-Asce 130(5): 616-625.

Kerzner, H. (2003). Project management : a systems approach to planning, scheduling, and controlling. Hoboken, NJ, Wiley.

Kolisch, R. (1999). "Resource allocation capabilities of commercial project management software packages." Interfaces 29(4): 19-31.

Kolisch, R. and S. Hartmann (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update." European Journal of Operational Research 174(1): 23-37.

Kolisch, R. and A. Sprecher (1997). "PSPLIB - A project scheduling problem library." European Journal of Operational Research 96(1): 205-216.

Kolisch, R., A. Sprecher, et al. (1995). "Characterization and generation of a general class of resource-constrained project scheduling problems." Management Science 41(10): 1693-1703.

Krishnan, V. (1996). "Managing the simultaneous execution of coupled phases in concurrent product development." Ieee Transactions on Engineering Management 43(2): 210-217.

- Krishnan, V., S. D. Eppinger, et al. (1997). "A model-based framework to overlap product development activities." Management Science 43(4): 437-451.
- Kusiak, A. (1993). Concurrent engineering : automation, tools, and techniques. New York ; Toronto, Wiley.
- Kusiak, A. and W. S. Chow (1987). "EFFICIENT SOLVING OF THE GROUP TECHNOLOGY PROBLEM." Journal of Manufacturing Systems 6(Compendex): 117-124.
- Liberatore, A. M., J. Schulz, et al. (2004). "2-alkyl-4-arylimidazoles: structurally novel sodium channel modulators." Bioorganic & Medicinal Chemistry Letters 14(13): 3521-3523.
- Lin, J., K. H. Chai, et al. (2009). "Optimal overlapping and functional interaction in product development." European Journal of Operational Research 196(3): 1158-1169.
- Liu, T., M. Liu, et al. (2005). "Hybrid genetic algorithm based on synthetical level of resource conflict for complex construction project scheduling problem." Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Vols 1-9: 5699-5703
- 5719.
- Loch, C. H. and C. Terwiesch (1998). "Communication and uncertainty in concurrent engineering." Management Science 44(8): 1032-1048.
- Pellerin, R. (1997). Modèle d'ordonnancement dynamique de projets de réfection. Montréal, Ecole Polytechnique de Montréal. Ph.D.
- Pritsker, A., L. Watters, et al. (1969). "Multi-project scheduling with limited resources: a zero-one programming approach." Management Science(16): 93-108.
- Project Management Institute (2008). Guide du corpus des connaissances en management de projet : guide PMBOK. Newtown Square, Penns., Project Management Institute.
- Roemer, T. A., R. Ahmadi, et al. (2000). "Time-cost trade-offs in overlapped product development." Operations Research 48(6): 858-865.
- Steward, D. V. (1981). "The Design Structure System: A Method for Managing the Design of Complex Systems." Ieee Transactions on Engineering Management EM-28(3): 71-74.

Terwiesch, C. and C. H. Loch (1999). "Measuring the effectiveness of overlapping development activities." Management Science 45(4): 455-465.

Wang, J. T. and Y. I. Lin (2009). "An overlapping process model to assess schedule risk for new product development." Computers & Industrial Engineering 57(2): 460-474.

ANNEXE 1 – Article soumis à « *International Journal of Project Organisation and Management* »

Evaluating the effectiveness of task overlapping as a risk response strategy in engineering projects

Autors : L. Grèze, R. Pellerin, P. Leclaire, N. Perrier

Abstract

In the last two decades, the failure of multiple engineering projects has highlighted the importance of adopting risk management practices. While risk identification and risk assessment have been widely studied in the literature, only few authors have proposed formal tools for helping project managers to evaluate the effectiveness of their risk response plan. While some risk response measures might be easily validated, overlapping, a commonly used mitigation measure in engineering projects is difficult to evaluate because of the complex interactions between activities and resources. This paper proposes an evaluation model to measure the effectiveness of overlapping strategy as a risk response in terms of additional cost and total maximum time reduction. Results based on a large set of generated projects highlight the importance of three factors in the effectiveness of an overlapping strategy: the number of opportunities of overlapping, the maximum overlapping amount allowed, and the level of resource constraints.

Keywords:

Project management, risk management, concurrent engineering, activity overlapping, scheduling

Introduction

Most project schedules are developed in a deterministic manner where activity durations correspond to a single value, usually the most likely duration. The assumption is that the duration is known with some certainty. However, the schedule often contains significant uncertainty, especially in complex technical environment and in new product development projects. In fact, these types of projects are usually composed of a large number of interrelated tasks. The complexity of the information flow as well as the possible dependency between tasks make the project scheduling difficult and unsteady. The potential need of repeating a certain task within the development process also worsens the situation (Chen *et al.* 2003).

In the last two decades, the failure of multiple engineering projects has highlighted the importance of adopting risk management practices (Lee *et al.* 2009; Williams, 1995). Project risk management aims at reducing risk by developing a project plan which minimizes the uncertainty and by implementing strategies which maximize the probability of achieving the project goals. Various studies have discussed means of conducting risk management by proposing formal processes for achieving project success (Cooper *et al.* 2005; Patterson and Neaily, 2002; Smith and Merrit, 2002; Chapman, 1997). While the proposed process and the terminology may differ from one author to another, the general risk management process consists of four main phases: risk identification, risk assessment, risk response planning, and risk control.

Among these phases, risk identification and risk assessment have been the most widely studied in the literature. For instance, several risk analysis tools or frameworks were proposed to identify and quantify the risks in new product development projects (Choi and Ahn 2010; Kavis *et al.* 2006). Most traditional risk assessment approaches result in probabilistic analysis.

On the other hand, only few authors have proposed formal tools for helping project managers to evaluate the effectiveness of their risk response plan, also called the risk mitigation plan. This plan, which may include various risk avoidance, risk reduction, and risk transfer measures, is however crucial in complex projects by formalizing the risk contingency strategy of the organization.

While some risk response measures might be easily validated, others might be more difficult to evaluate in engineering projects. Engineering projects involve many variables and it is often difficult to determine dependence and correlations between them. For instance, overlapping is a core technique for reducing development time (Bogus *et al.*, 2005; Terwiesch *et al.*, 1999; Smith *et al.*, 1995; Smith and Reinertsen 1998) and is widely used as a risk response strategy when anticipating development delay in the early phases of projects. Overlapping consists in starting an activity before receiving all the final information required. Its efficiency for reducing product development time has been proved in the aerospace (Sabbagh 1996) and automobile industries (Clark and Fujimoto 1991). However, this practice often causes future rework and modification as new information is gained in subsequent activities. In some cases, rework may outweigh the benefits of executing activities in parallel (Terwiesch *et al.*, 1999). The real benefits of activity overlapping also largely depend on the nature of the project schedule as overlapping creates interactions between activities. As such, the project topology, as defined by its network and the resource constraints may greatly affect the effectiveness of overlapping measure. Consequently, the total expected reduction of time is difficult to evaluate and additional costs associated with rework are often ignored in the project planning phases. It is therefore not surprising to note that most companies determine overlapping strategies on an ad hoc basis without always considering rework (Lin *et al.*, 2009). Most contributions in the literature also failed to consider the impact on resources when planning overlapping development activities or are limited to measuring the project time reduction obtained through overlapping a posteriori by analyzing completed projects (Terwiesch and Loch, 1999).

In this paper, we propose a different approach by presenting a predictive model taking into account the interactions between activities and resource constraints. Our objective is to propose a contingency plan evaluation model to measure the effectiveness of overlapping strategy in terms of additional cost and total maximum time reduction. This

paper is based on a deterministic resource scheduling model which determines the optimal sets of overlappable activities and modes. The calculated project makespan improvement can then be used as an upper bound which defines the maximum gain that can be obtained by accelerating development activities through overlapping. These results allow project managers to determine if overlapping is an adequate risk reduction measure and what are the additional budget requirements associated with that strategy if implemented during the project execution.

The remainder of the paper is organized as follows. Section 2 first presents a brief review of overlapping execution studies. Section 3 describes the proposed solution approach followed by the description of the experimental data set used in this study in Section 4. Computational results are then presented in section 5. The paper concludes with recommendations for future work in section 6.

Literature review

Overlapping depends not only on dependency between activities but also on information exchange policy between upstream and downstream activities and progress evolution. Two groups of models have been developed in the literature to analyze overlapping interactions. First, many authors consider only couples of activities and no resource constraint to establish the best trade-off between overlapping and rework. For instance, Krishnan *et al.* (1997) developed a model-based framework to manage the overlapping of coupled activities. This model introduces the concept of information evolution and downstream sensitivity to describe interaction between both activities. Information evolution refers to the upstream generated information useful for downstream activities. Downstream sensitivity refers to the impact of a change in upstream activity on the downstream activity. The more significant the impact is, the higher the sensitivity is.

Relying on these concepts, Krishnan (Krishnan, 1996) defined different types of appropriated overlapping strategies: iterative, preemptive, distributive and divisive overlapping. In a similar manner, Bogus *et al.* (2006) identified appropriate strategies to efficiently implement overlapping in practice. Lin *et al.* (2009) also improved the overlapping model by incorporating the downstream progress evolution and determined the optimal overlap amount. These models assume that overlapping parameters can be derived from historical data of projects.

Other approaches have considered whole projects instead of couples of activities under the assumption that relation between overlap amount and rework is preliminary known for overlappable activities. They mostly use design structure matrix (DSM) to represent dependencies, to minimize feedbacks, and to identify overlapping opportunities between activities. DSMs were introduced by Steward (1981). Among these models, Gerk and Qassim (2008) developed an analytic project acceleration linear model via activity crashing, overlapping and substitution with resource constraints. Wang and Lin (2009) developed a stochastic overlapping process model to assess schedule risks. Their simulation model considers iterations and probabilities of rework. Iterations are mostly defined as interaction between design activities which lead to rework caused by feedbacks from downstream activities.

However, these models do not take into account resource constraints, except for Cho and Eppinger (2005) who introduced a simulation model with stochastic activity durations, overlapping, iterations, rework, and resource constraints for some activities. They showed that these constraints can delay some overlapped activities and delay the project. All these models assume a simple linear relationship between rework and overlap amount with an upper bound and a lower bound.

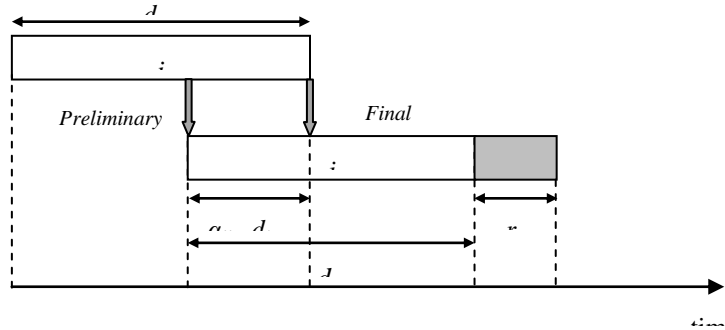
The proposed approach

A project is defined by a set of n activities, including two fictitious activities 0 and $n+1$, which correspond to the project start and project end, with zero processing time. We denote by d_j the nominal processing time of activity j considering that all the final information required from preceding activities are available at its start; in other words, if activity j is processed without overlapping.

It is here assumed that a project is only composed of independent and dependent activities. The resulting information flow within the project between activities is assumed to be unidirectional from upstream to downstream activities. The analysis of information exchanges between dependent coupled activities enables to categorize them into non-overlappable and overlappable ones. The former represents the case where a downstream activity requires the final output information from an upstream activity to be executed or the completion of the upstream activity. The latter represents the case where a downstream activity can begin with preliminary information and receives final update at the end of the upstream activity. This relation provides the opportunity to overlap two activities so that a downstream activity can start before an upstream activity is finished. While the non-overlappable activities are connected with the classical finish-to-start precedence constraint, the overlappable ones are connected with a finish-to-start-plus-lead-time precedence constraint where the lead-time accounts for the amount of overlap (Cho and Eppinger, 2005).

Figure 1 shows the overlapping process of an overlappable couple of activities i and j . The downstream activity j starts with preliminary inputs from the upstream activity i . The amount of overlap, α_{ij} , is expressed as a fraction of the downstream activity duration. An additional rework is often necessary to accommodate the changes in the upstream information in the downstream development. The expected duration of this rework is denoted by r_{ij} .

Figure 1: Overlapping process of two activities



An important part of the literature on overlapping process is dedicated to the determination of the optimal overlap amount for a couple of activities without resource constraint (Krishnan *et al.*, 1997; Roemer *et al.*, 2000; Terwiesch and Loch, 1999). However, the optimal overlap amounts for a resource-constrained project composed of several couples of overlappable activities are not necessarily set to the optimal values found for each couple of activities (Cho and Eppinger, 2005; Gerk and Qassim, 2008; Browning and Eppinger, 2002).

To determine the optimal schedule when overlapping is allowed, we propose a linear 0-1 integer programming model which is based on a previous model developed by Berthaut *et al.* (2011). Within this resource constraint scheduling model, presented in annex, overlapping is supposed to be performed for discrete values of overlap amount. In practice, however, scheduling is performed on a period by period basis and activity progress is measured according to the completion of milestones, corresponding to activity deliverables. Therefore, we consider a finite number of different overlap amounts aligned with activity milestones. These different values constitute the different feasible modes. Each mode is characterized by a specific amount of overlap and associated rework duration.

In order to represent the impact of overlapping on the execution cost, the overlapping costs is assumed to be composed of the cost of rework and the additional coordination cost. The coordination cost is considered negligible, which is also the case in previous works (Gerk and Qassim, 2008; Krishnan *et al.*, 1997). It is assumed that rework cost is considered as a linear function of the time spent on rework, where the linear factor is the average wages of the teams per unit of time. It is also assumed that all resources have the same period rate (*i.e.* the resource use cost is estimated at \$100 per period for all resources).

Experimental results are presented in the next section.

Experimental data

In order to demonstrate the importance of project characteristics, and the necessity to consider resource constraints when evaluating the maximum time reduction that can be obtained by adopting overlapping measures, we tested the proposed approach with different projects having similar characteristics.

A reference project consisting of 30 activities was used to build our experimental data set. The PROGEN project generator (Kolish *et al.*, 1992) was used to generate comparable project networks and resource demand. All projects involved 4 renewable resources. As in Kolish *et al.* (1992), we use three main parameters to define our project data set: the project complexity C , the resource factor RF , and the resource strength factor RS . The complexity parameter is defined as the average number of arcs per node. This parameter describes the complexity in relations and dependences between activities and is used to construct the activity-on-node project networks. The resource factor is calculated as the average portion of requested resources per activity. If $RF=1$, all 4 resources will be needed for each activity. Instead, if $RF=0.5$, then each activity will require in average 2 different types of resources. The quantity of each requested resource is randomly calculated between the minimum and maximum allowed values. Finally, the resource strength factor characterizes the relation between resource demand and resource availability. Kolish *et al.* (1992) present a methodology to determine resource availability with this RS factor:

$$Q_k = Q_{k,min} + Round(RS * (Q_{k,max} - Q_{k,min}))$$

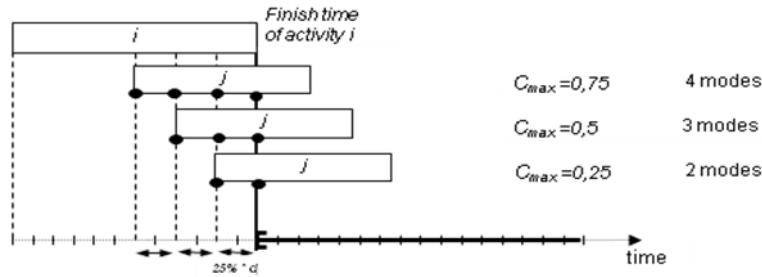
where $Q_{k,min}$ is the minimum acceptable value for the resource k availability to complete the project, equal to the maximum resource k request per job. $Q_{k,max}$ is calculated as the peak demand of resource k in the initial makespan, without resource constraint and with earliest start schedule. Q_k is the resource k availability during the whole project. The RS factor therefore represents the influence of resource constraints on the project.

In our design of experiment, we fixed a median value for C and RF , 2.1 and 0.5 respectively, and generated 9 projects with similar network characteristics and resource demand. For each project, three different characteristics of resource availability were created ($RS=1$, 0.75 and 0.5). Finally, 27 comparable projects without overlap data are generated (9 for each value of RS).

Three additional parameters were considered to generate overlap data. First, we set the factor R the percentage of overlappable couples of activities among all couples of dependent activities. This factor illustrates opportunity of overlapping, and reflects a strategy of project execution. Indeed, a project can allow a more or less important number of couples of overlappable activities depending on the nature of the activities and strong or flexible dependency relationship between them. In the literature, some authors present many cases of overlapping situations, considering from 100% to 10% of overlappable activities among all couples of activities (Gerk and Qassim, 2008). In order to present different cases encountered in practice, three different values of R , 20%, 40% and 60% we here considered.

The second factor C_{max} is the maximum allowable amount of overlap. The value of this factor is determined by activity milestones and previous information needed to start the downstream activity. All overlappable activities were assumed to have similar milestones and the same maximum amount of overlap. In the literature, authors have used various maximum of overlap (Gerk and Qassim 2008; Wang and Lin, 2009). In this paper, three different values of C_{max} were considered: 25%, 50% and 75%, which correspond to a conservative, median and aggressive possibility of overlapping execution, respectively. Figure 2 illustrates the different modes associated to these values.

Figure 2: Possible values of C_{max} and associated modes



The third factor β is the amount of rework per period of overlapping. This amount depends on the nature of the project and its associated risks. This amount, set to a value of 40%, is here considered to be the same for all activities. In practice, the project planner has to determine the amount of rework for each overlappable activity before using an overlapping strategy, but our assumption allows to obtain an upper bound which defines the maximum gain that can be obtained by accelerating development activities through overlapping, and the maximum induced cost.

The three factors, R , C_{max} and β were used to generate overlapping data for each project. Overlappable couples of dependent activities are randomly selected in the list of dependent activities. An opportunity list is established with the highest value of R . Then, for each value of R , a choice is made randomly in this list of couples of activities. The percentage of overlappable couples is equal to the defined value of R . Once the overlappable couples are identified, the overlap mode data is generated, with overlap amount per mode and associated rework.

Table 1 summarizes our design of experiment. The resource strength factor RS , the percentage of overlappable couples of activities among the total couples of dependent activities and the maximum amount of overlapping C_{max} , were all tested at three levels on 9 similar project topologies generated with PROGEN, resulting in 243 projects.

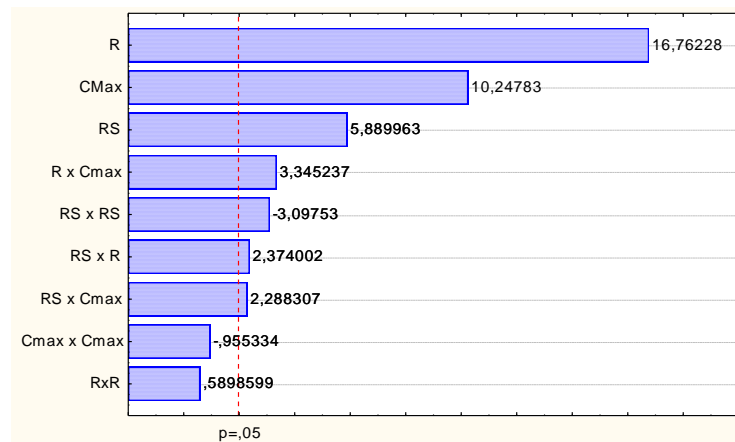
Table 1: Variable parameter levels

RS	0.5	0.75	1
R	0.2	0.4	0.6
C_{max}	0.25	0.5	0.75

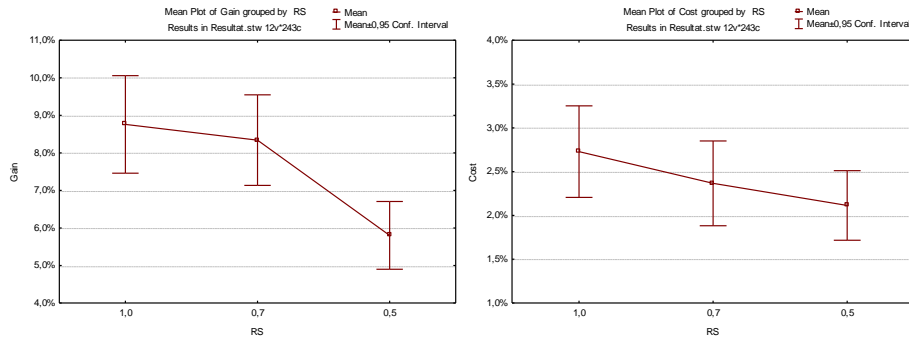
Computational results

The 243 generated projects were implemented in AMPL Studio v1.6.j and solved with CPLEX 12.2. Because of overlapping opportunities defined in our data generation procedure, the results in terms of optimal time reduction vary for each project. As each project has a different initial makespan, the effectiveness of overlapping strategy was measured as the percentage of time reduction when comparing the calculated project makespan with overlap with the initial project schedule. Additional costs resulting from rework in also measured in a similar manner. The resource cost is estimated at \$100 per period and we considered a \$20,000 fixed cost.

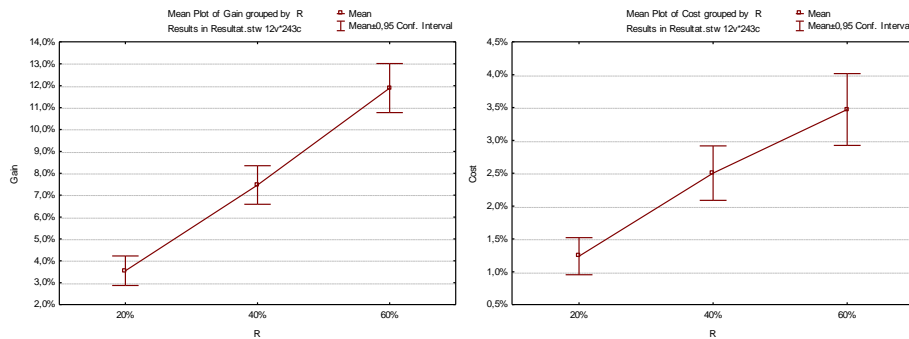
The effects of the different parameters were analyzed by conducting ANOVA. The resulting Pareto chart, presented in figure 3, indicates that the three factors explain 80% of variation in effectiveness of overlapping in terms of time reduction and cost increase. The chosen factors thus have a significant impact in overlapping effectiveness. We also observed important interaction between the percentage of overlapping opportunities R and the maximum allowed amount of overlapping and between the availability of resources and the two other factors. These interactions clearly show the indirect impact of resource constraints on the effectiveness of overlapping as a risk response strategy.

Figure 3: Pareto chart of standardized effects

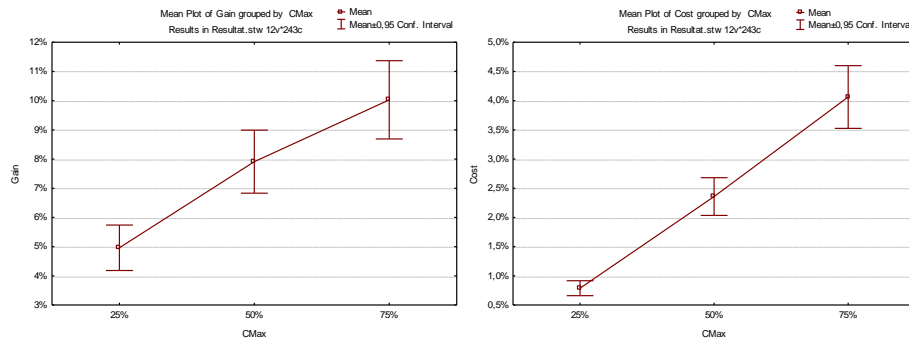
From figure 4, one can see the mean effect of RS in the effectiveness of overlapping strategy as a risk response in terms of additional cost and total maximum time reduction. RS has a strong impact on effectiveness but is not significant in terms of cost influence. Indeed, the reduction of RS decreases the opportunity of overlapping and the resulting time gain. This is due to strong resource constraints reducing overlapping opportunities. The large gap between $RS=0.75$ and $RS=0.5$, where the gain varies from 8.8% to 5.8% shows the importance of resource constraints and the necessity to take them into account when scheduling projects. Indeed, overlapping increase the impact of resource constraints as concurrent activities may require the same resources at the same time. The time of rework has also an impact on resource constraints as every overlapped activity requires additional labor time due to the addition of rework.

Figure 4: *RS* impact in terms of time reduction and cost

The impact of altering the number of overlapping opportunities *R* can be seen in figure 5. As *R* increases, the effectiveness of overlapping increases too, from 2.5% to almost 12%. This is due to the numerous overlapping opportunities which allow to bypass many resource constraints and finally to decrease the time of execution of the project. The impact on cost is concave: the greater the number of overlappable couples of activities is, the less important the influence of *R* on cost is. This can be explained by the increasing number of overlap opportunities which allows small overlapping to be efficient. This point highlights the significance of identifying correctly the couples of overlappable activities in the risk response planning phase.

Figure 5: *R* impact in terms of time reduction and cost

The increase in the maximum overlapping factor results in a concave increase of gained time (*cf.* figure 6). Significant overlapping modes allow to reduce effectively the time of execution of the project. However, resource constraints tend to reduce the possibilities of major overlapping. On the contrary, the impact on cost seems to be linear from 0.8% to 4%. The maximum overlap amount allowed has a direct impact on rework time.

Figure 6: C_{max} impact in terms of time reduction and cost

These computational results show that a high resource level combined with numerous opportunities of overlapping and a high percentage of maximum overlapping favor overlapping to be an efficient risk response strategy when anticipating project delay. The most important factor affecting the effectiveness of overlapping as a risk response strategy is the number of overlappable couples of activities. Resource availability has less direct impact but this factor has a high interaction with the other factors. Finally, the maximum amount of overlapping, as well as its interaction with the number of overlappable couples, is important to be considered when evaluating the effectiveness of overlapping measures.

The most important interaction, between the number of overlappable couples and the maximum amount of overlapping, were further analyzed by studying three scenarios. The first one is a project with many overlappable couples of activities and a low amount of maximum overlapping. The second one is a median one in terms of opportunities of overlapping and of maximum amount of overlapping. The third one is a strategy with few overlappable couples and high amount of maximum overlapping. As shown in Table 2, the first and second strategies are more effective than the third one but the median strategy results in additional cost. This result supports the fact that the number of overlappable activities greatly affects the effectiveness of an overlapping strategy.

Table 2: Mean impact of project characteristics in overlapping effectiveness

R	C_{max}	Time reduction	Cost increase
60%	25%	8.4%	1.1%
40%	50%	8.3%	2.4%
20%	75%	4.6%	2.0%

Conclusion

This study shows the influence of three main factors influencing the effectiveness of adopting overlapping as a risk mitigation measure when anticipating project delay in early phases of engineering projects. The percentage of overlappable activities is a predominant factor of effectiveness while having a low impact on additional cost associated with rework. Overlapping effectiveness increases with number of overlappable activities by creating more opportunities to resolve resource constraints, and therefore reducing the total project duration.

The maximum amount of overlapping allowed also impacts the effectiveness of overlapping measures. However, the increase in project time reduction comes with the expense of additional work and costs. Allowing important amount of overlapping must be studied carefully as it may create additional risks for the project.

The impact of resource constraints on the maximum time reduction that can be obtained through overlapping measure is another important result of this study. First, our experimental results show that strong resource constraints greatly reduce the possibilities of overlapping as concurrent activities may require the same resources and thus, creating resource conflicts that can be only resolved by delaying some activities. Consequently, overlapping activities may in some cases have no impact in total project duration. In addition, our results demonstrate that the potential benefits of overlapping measures vary greatly from one project to another. Even when comparing similar projects, the maximum gain in time reduction depends on the characteristics of the initial project schedule. This result demonstrates the usefulness of the proposed approach which relies on a project scheduling model.

We can also conclude that defining risk response strategies relying solely on historical data from previous projects is not sufficient to adopt an adequate risk response plan. Risk mitigation measures must be assessed by analyzing the impact of project resources. Historical data are however useful by providing insights when determining the maximum amount of overlap that should be allowed and by estimating the amount of rework associated with overlapping.

We would like also to point out some limitations of the proposed approach and suggest possible directions for future research. First, our approach relies on an optimal scheduling model. As in most resource scheduling approaches, computation time is an issue when analyzing real projects. In our experiments, the number of overlapping modes has an important impact on computational time, tending to make impossible the implementation of this upper bound analysis in practice. Future works could develop heuristics or meta-heuristics to compare the effectiveness of different risk response strategy as overlapping.

Secondly, we considered the impact of overlapping on total project duration and assumed that the inherent risks of this measure can be simply taking into accounts by considering rework and its related cost. In practice, adopting overlapping may face other risk elements, as exposed in the concurrent engineering literature (Kays et al., 2006). These elements must be analyzed as well.

Finally, our model calculates the maximum gain of overlapping measures from a discrete point of view. A stochastic approach could offer a satisfactory estimation of effectiveness of overlapping, rather than an upper bound. We have also to mention that the calculated upper bound is only feasible at the beginning of the project. As the project progress, this maximum gain decreases. A project rescheduling approach should be adopted within the risk control activities to continuously assess the effectiveness of overlapping as a valid measure to respond to project delay.

References

- Berthaut, F, Grèze L, Pellerin, R, Perrier, N & Hajji A 2011, 'Optimal resource-constraint project scheduling with overlapping modes', submitted for *International Conference on Industrial Engineering and Systems Management*.
- Bogus, SM, Molenaar, KR & Diekmann, JE 2005, 'Concurrent engineering approach to reducing design delivery time', *Journal of Construction Engineering and Management*, vol. 131, no. 11, pp. 1179–1185.
- Bogus, SM, Molenaar, KR & Diekmann, JE 2006, 'Strategies for overlapping dependent design activities', *Construction Management and Economics*, vol. 24, pp. 829–837.
- Browning, TR & Eppinger, SD 2002, 'Modeling impacts of process architecture on cost and schedule risk in product development', *IEEE Transactions on engineering management*, vol. 49, no. 4, pp. 428–442.
- Chapman, C 1997, 'Project risk analysis and management – PRAM the generic process', *International Journal of Project Management*, vol. 15, no. 5, pp. 273–281.
- Chen, CH, Ling, SF & Chen, W 2003, 'Project scheduling for collaborative product development using DSM', *International Journal of Project Management*, vol. 21, pp. 291–299.
- Cho, SH & Eppinger, SD 2005, 'A simulation-based process model for managing complex design projects', *IEEE Transactions on engineering management*, vol. 52, no. 3, pp. 316–328.
- Choi, HG & Ahn, J 2010, 'Risk analysis models and risk degree determination in new product development: A case study', *Journal of Engineering and Technology Management*, vol. 27, pp. 110–124.
- Clark, KB & Fujimoto, T 1991, *Product development performance strategy, organization and management in the world auto industry*, Harvard Business School Press, Boston, MA.
- Cooper, DF, Greay, S, Raymon, G & Waler, P 2005, *Project risk management guidelines: Managing risk in large projects and complex procurements*, Jown Wiley and Sons, Ltd.
- Gerk, JEV & Qassim, RY 2008, 'Project acceleration via activity crashing, overlapping, and substitution', *IEEE Transactions on Engineering Management*, vol. 55, no. 4, pp. 590–601.
- Kayis, B, Arndt, G, Zhou, M, Savci, S, Khoo, YB & Rispler, A 2006, 'Risk Quantification for New Product Design and Development in a Concurrent Engineering Environment', *CIRP Annals - Manufacturing Technology*, vol. 55, no. 1, pp. 147–150.
- Kolisch, R, Sprecher, A & Drexel, A 1992, 'Characterization and generation of a general class of resource-constrained project scheduling problems - Easy and hard instances', Research Report 301, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel
- Krishnan, V 1996, 'Managing the simultaneous execution of coupled phases inconcurrent product development', *IEEE Transactions on Engineering Management*, vol. 43, no. (2), pp. 210–217
- Krishnan, V, Eppinger SD & Whitney DE 1997, 'A model-based framework to overlap product development activities', *Management Science*, vol. 43, no. 4, pp.437–451.
- Lin, J, Chai, KH, Brombacher, AC & Wong, YS 2009, 'Optimal overlapping and functional interaction in product development', *European Journal of Operational Research*, vol. 196, no. 3, pp. 1158–1169.
- Lee, E, Park, Y, & Shin, JG 2009, 'Large engineering project risk management using a Bayesian belief network', *Expert Systems with Applications*, vol. 36, pp. 5880–5887.

- Patterson, FD & Neailey, K 2002, 'A risk register database system to aid the management of project risk', *International Journal of Project Management*, vol. 20, no. 5, pp. 365–374.
- Roemer, TA, Ahmadi, R & Wang, RH 2000, 'Time-cost trade-offs in overlapped product development', *Operations Research*, vol. 48, no. 6, pp. 858–865.
- Sabbagh, K 1996, *Twenty-first century jet*, Scribner, New York.
- Smith, PG & Merritt, GM 2002, *Proactive Risk Management: Controlling Uncertainty in Product Development*, Productivity Press, New York.
- Smith, P & Reinertsen, D 1995, *Developing Products in Half the Time*, New York: Van Nostrand Reinhold.
- Smith, PG & Reinertsen, DG 1998, *Developing Products in Half the Time*, 2nd edn. John Wiley, New York.
- Steward, DV 1981, 'The design structure system: a method for managing the design of complex systems', *IEEE Transactions on Engineering Management*, vol. 28, pp. 71–74.
- Terwiesch, C & Loch, CH 1999, 'Measuring the effectiveness of overlapping development activities', *Management Science*, vol. 45, no. 4, pp. 455–465.
- Wang, JT & Lin, YI 2009, 'An overlapping process model to assess schedule risk for new product development', *Computers & Industrial Engineering*, vol. 57, no. 2, pp. 460–474.
- Williams, T 1995, 'A classified bibliography of recent research relating to project risk management', *European Journal of Operational Research*, vol. 85, no. 1, pp. 18–38.

Annex: The linear 0-1 integer program

Chapitre 6 Table 1: Symbols and definitions

Chapitre 7

Symbol	Definition
S	Set of activities
n	Number of non-dummy activities
$E = A \cup P$	Set of temporal or precedence constraints
$i \rightarrow j (i, j)$	Precedence constraint
d_j	Processing time of activity j
A	Set of couples of overlappable activities
P	Set of couples of non-overlappable activities
$A(j)$	Set of immediate predecessors of activity j that are overlappable with activity j
$P(j)$	Set of immediate predecessors of activity j that are not overlappable with activity j
$Pred(j) = A(j) \cup P(j) \forall j \in S$	Set of immediate predecessors of activity j
R	Set of renewable resources
R_k	Constant amount of available units of renewable resource k
R_{jk}	Per period usage of activity j of renewable resource k
m_j	Number of execution modes of activity j
α_{ijm}	Amount of overlap duration between activities i and j in execution mode m , expressed as a fraction of d_j
r_{jm}	Expected amount of rework in activity j in execution mode m
T	Upper bound of the project's makespan
$t = 0, \dots, T$	Periods
EF_j	Earliest possible finish time of activity j
LF_j	Latest possible finish time of activity j

Each activity j must finish within the time window $\{EF_j, \dots, LF_j\}$ with respect to the precedence relations and the activity durations. They can be derived from the traditional forward recursion and backward recursion algorithms considering that the project must start at time 0 and that T constitutes an upper bound of the project makespan (*i.e.* the sum of processing times of all activities) (Hartmann, 1999). We define the decision variables (*i.e.* the finish times and the overlapping modes) as follows:

$$X_{jtm} = \begin{cases} 1 & \text{if activity } j \text{ is executed in mode } m \text{ and finished at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$\forall j \in S, \forall t \in [0, T] \text{ and } \forall m \in [1, m_j] \quad (1)$$

The decision on the activity modes can be classed into three cases. On the one hand, if activities (i, j) are not overlappable, the decision is simply not to overlap. On the other hand, if activities (i, j) are overlappable, these activities can be either overlapped ($m > 1$) or executed in series ($m=1$). The resource-constrained scheduling problem with overlapping can then be formulated as a linear 0-1 integer program as follows:

$$\text{Minimize } \sum_{m=1}^{m_{n+1}} \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,m} \quad (2)$$

Subject to:

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} (t - d_j \cdot (1 - \alpha_{ijm}) - r_{jm}) \cdot X_{jtm} \quad \forall j \in S, \forall i \in \text{Pred}(j) \quad (3)$$

$$\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} \alpha_{ijm} \cdot X_{jtm} \leq Y_{ij} \quad \forall j \in S, \forall i \in A(j) \quad (4)$$

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \geq \left(\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} (t - d_j \cdot (1 - \alpha_{ijm}) - r_{jm}) \cdot X_{jtm} \right) - T \cdot (1 - Y_{ij}) \quad \forall j \in S, \forall i \in A(j) \quad (5)$$

$$\sum_{j=2}^n \left[R_{jk} \cdot \left(\sum_{m=1}^{m_j} \sum_{b=t}^{t+d_j-1+r_{jm}} X_{jbm} \right) \right] \leq R_k \quad \forall k \in R \quad \forall t \in [1, T] \quad (6)$$

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} t \cdot X_{jtm} \quad \forall j \in S, \forall i \in A(j) \quad (7)$$

$$\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} X_{jtm} = 1 \quad \forall j \in S \quad (8)$$

$$X_{jtm} \in \{0,1\} \quad \forall j \in S, \forall t \in [0, T] \text{ and } \forall m \in [1, m_j] \quad (9)$$

$$Y_{ij} \in \{0,1\} \quad \forall j \in S, \forall i \in \text{Pred}(j) \quad (10)$$

The objective (2) minimizes the finish time of the dummy sink activity and therefore, the project makespan. Constraint (3) represents the finish-to-start precedence constraints, with a negative lead time in the case of overlapping. According to constraints (4), if two overlappable activities (i,j) are overlapped, then $Y_{ij} = 1$. If activities (i,j) are not overlapped, then Y_{ij} is unrestricted and constraints (5) are not restrictive.

Constraints (6) define the resource constraints. Constraints (7) guarantee that the downstream activity of a couple of overlappable activities can not finish before the upstream activity finish time. Constraints (8) ensure that each activity is assigned one activity mode and one finish time. Finally, constraints (9) and (10) define the aforementioned binary decision variables.

ANNEXE 2 – Article soumis pour la conférence du CIGI

CIGI 2011

Processus d'accélération de projets sous contraintes de ressources avec modes de chevauchement

LUCAS GREZE¹, ROBERT PELLERIN¹, PATRICE LECLAIRE²¹ CHAIRE DE RECHERCHE JARISLOWSKY/SNC-LAVALIN EN GESTION DE PROJETS INTERNATIONAUX

Ecole Polytechnique de Montréal

C.P. 6079, succ. Centre-ville

Montréal (QC), Canada H3C3A7

lucas.greze@polymtl.carobert.pellerin@polymtl.ca² INSTITUT SUPÉRIEUR DE MÉCANIQUE DE PARIS –SUPMÉCA

3 rue Fernand Hainaut, 93407 Saint-Ouen, France

patrice.leclaire@supmeca.fr

Résumé –

Le chevauchement d'activités est une pratique courante pour accélérer l'exécution de projets. Cette technique consiste à exécuter en parallèle deux activités, normalement séquentielles, en autorisant l'activité en aval à commencer avec des informations préliminaires. Dans la pratique, le chevauchement est étudié en prenant compte les contraintes de ressources, mais ne prend pas toujours en compte les interactions entre activités, qui peuvent entraîner des retouches et des augmentations de coût. À l'opposé, la littérature scientifique comporte beaucoup d'études sur les interactions qui ne prennent pas en compte les contraintes de ressources. Dans ce papier, nous proposons un processus d'accélération de projet sous contrainte de ressources à l'aide du chevauchement d'activités. Le processus propose l'identification des couples d'activités chevauchables, l'évaluation des paramètres de chevauchement, jusqu'à l'ordonnancement du projet. Nous considérons ici deux techniques d'ordonnancement, soit un modèle de programmation linéaire et un modèle heuristique. Ces deux approches ont été testées sur neuf instances de projets pour illustrer et quantifier les gains obtenus. Les résultats obtenus illustrent l'efficacité du chevauchement en tant que technique d'accélération de projet. Nous présenterons aussi les limites d'une méthode de résolution exacte pour une approche industrielle et l'efficacité des méthodes heuristiques.

Abstract –

Overlapping activities is widely used to accelerate project execution. Overlapping consists in executing in parallel two sequential activities by allowing a downstream activity to start before the end of an upstream activity based on preliminary information. In companies, overlapping is examined in resource constraints scheduling without considering interaction between activities and rework caused by alteration of information exchanged until finalized information is available at the completion of the upstream activity. By contrast, the literature deals with overlapping of couple of activities without considering a whole project with resource constraints. We propose a process of accelerating project using the overlapping of activities. This process describes the identification of overlappable couples of activities, the assessment of overlap parameters and then the scheduling of the project. We propose in this paper two different scheduling approaches, an integer linear programming problem and a heuristic algorithm. We applied this process on nine instances of project to illustrate and quantify the associated gains. Results illustrate the effectiveness of overlapping as a technique for accelerating project execution. We also present the limitations of an exact resolution of the RCPSP problem.

Mots clés - Chevauchement d'activité, ingénierie simultanée, gestion de projet, ordonnancement de projet**Keywords – Activity overlapping, concurrent engineering, project management, project scheduling**

Introduction

Les problèmes d'ordonnancement de projet avec contraintes de ressources (en anglais, RCPSP, *Resource Constrained Project Scheduling Problems*), constituent un domaine d'étude de prédilection. Parmi les extensions étudiées dans la littérature, différentes pratiques, dont le chevauchement, la compression et la substitution d'activités [Gerk et Quassim, 2008] ont été mises en avant pour réduire le temps d'exécution du projet afin de produire un échéancier initial ou pour le modifier en cours d'exécution. Parmi les techniques d'accélération préconisées lors de projets d'ingénierie, le chevauchement d'activités s'avère être une technique privilégiée pour réduire le temps nécessaire aux phases de développement [Bogus et al., 2005]. En pratique, cette technique consiste à démarrer une activité avant de recevoir l'ensemble des informations requises au préalable. Cela implique toutefois des tâches de retouche et des modifications suite à la réception d'informations complémentaires transmises après le démarrage de l'activité. Cette pratique génère dans bien des cas une charge de travail et un coût d'exécution supplémentaires. Au final, il en résulte une réduction du temps total d'exécution du projet, représentée par la différence entre la période de chevauchement et le temps supplémentaire du aux retouches, et un coût supplémentaire associé.

Parmi les études portant sur le sujet, on note deux types de modèles de chevauchement développés dans la littérature. Le premier type considère uniquement des couples d'activités sans contrainte de ressources pour établir le meilleur compromis entre chevauchement et retouches. Krishnan et al. [Krishnan et al., 1997] ont ainsi introduit les concepts d'évolution de l'information et de sensibilité de l'activité aval pour décrire les interactions entre les activités. L'évolution décrit l'évolution de l'information utile pour l'activité en aval générée par l'activité en amont. La sensibilité mesure l'impact sur l'activité en aval des changements au sein de l'activité en amont. Plus l'impact est important, plus la sensibilité est grande. A partir de ces concepts, Krishnan a défini différentes stratégies de chevauchement adaptées.

Le second type de modèle de chevauchement considère, non pas un couple d'activités, mais un projet dans sa globalité en supposant que la relation entre la durée de chevauchement et la durée de retouche nécessaire est connue préalablement pour les activités chevauchables [Lin et al., 2009]. La plupart de ces modèles utilisent des matrices de décomposition ou DSM (*Design Structure Matrix*), introduites par Steward [Steward, 1981] pour représenter les dépendances entre les activités, limiter les rétroactions et identifier les opportunités de chevauchement d'activités [Maheswari et Varghese, 2005]. Parmi les modèles développés, Gerk et Quassim ont proposé un modèle linéaire analytique d'accélération de projet utilisant la compression d'activité, le chevauchement et la substitution et en considérant des contraintes de ressources. Wang et Lin [Wang et Lin, 2008] ont aussi développé un modèle de simulation considérant des itérations et des probabilités de retouche.

Parmi ces approches, aucune ne considère des possibilités réalistes de chevauchement. En effet, toutes les configurations de chevauchement ne sont pas acceptables dans la réalité. Elles

doivent correspondre à un certain mode d'exécution du couple d'activités qui correspond avec un échange d'information. L'objectif premier de ce papier est de proposer un processus d'accélération de projet avec contraintes de ressources à l'aide du chevauchement d'activités, applicable dans un milieu industriel. Nous proposerons donc dans un premier temps une méthode permettant d'obtenir des paramètres de chevauchement acceptables. Puis dans un second temps nous proposerons différentes méthodes de résolution du problème d'ordonnancement dans ce contexte.

La suite du papier est organisée de la façon suivante. La section 2 décrit la méthodologie proposée pour le processus d'accélération de projet sous contraintes de ressources avec mode de chevauchement. La section 3 présentera le modèle de programmation linéaire, les cas d'études ainsi que les résultats associés. La section 4 présentera l'approche heuristique et les résultats obtenus. Nous concluons le papier par une comparaison des différentes approches et une conclusion dans la section 5.

Méthodologie proposée

La figure 1 illustre la méthodologie que nous proposons. Les premières étapes du processus consistent à identifier les opportunités de chevauchement. Une fois ces opportunités identifiées, il faut choisir des modes de chevauchement pertinents et déterminer les degrés de chevauchement et les temps de retouches applicables à ces modes. Les modes de chevauchement décrivent des possibilités discrètes de chevauchement. La détermination de ces paramètres est cruciale pour l'application du modèle par la suite.

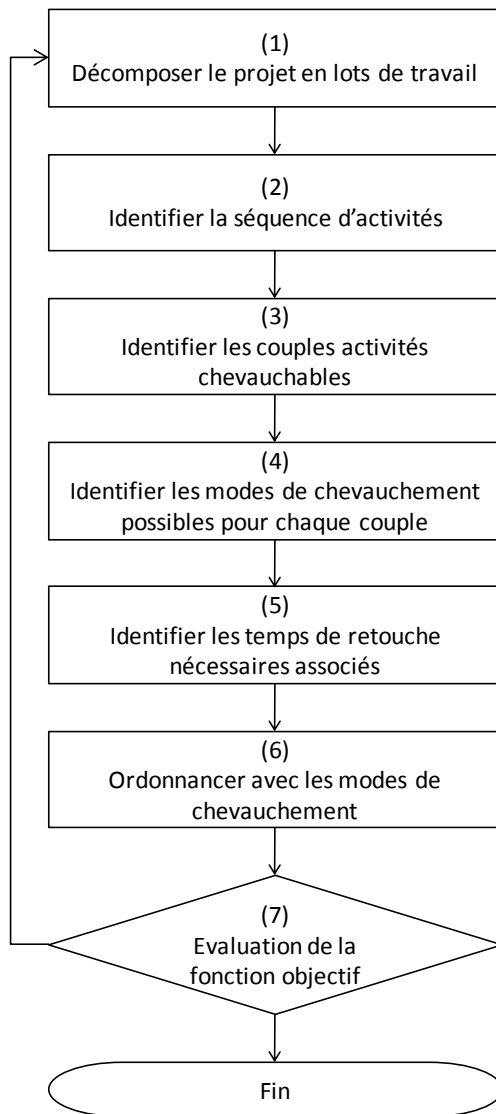


Figure 1 : Méthodologie d'accélération de projet

En premier lieu, il s'agit de décomposer le projet en lots de travail. La taille des lots de travail considérée dépend du niveau de détail exigé par la gestion du projet. Les lots de travail doivent représenter des activités de complexité et de durée comparables. C'est à partir de ces activités que nous allons considérer les possibilités de chevauchement.

Dans la pratique, c'est l'étude des interactions entre les activités qui permet d'identifier les activités chevauchables. La plupart des méthodes de planification de projet décrit les projets sous forme de réseaux, les nœuds représentant les activités (réseau «Activity on Node») ou les relations entre les activités (réseau «Activity on Arc»). Ces réseaux décrivent la séquence d'exécution des activités. Cependant, ces réseaux ne permettent pas de décrire les interactions entre les activités et de modéliser les flux d'information entre les activités.

Les matrices de décomposition (en anglais, *Design Structure Matrix*, DSM) permettent de modéliser ces relations

supplémentaires entre les activités [Browning, 2001]. Les échanges d'information entre activités peuvent avoir lieu à différents moments de l'avancement de l'activité. Il peut s'agir de plans, de données de dimensionnement ou de commandes de matériel, par exemple, nécessaires à l'exécution de l'activité en aval. Une matrice de décomposition est une matrice carrée où les lignes et les colonnes sont toutes deux associées aux activités. Les DSM visent à représenter les flux d'information entre les différentes activités, et permettent ainsi de repérer les retours d'information. Ces retours d'information sont compliqués à gérer au sein d'un projet car ils impliquent des changements dans une activité en cours ou terminée [Wang et Lin, 2008]. Les retours en arrière causent donc des modifications et du travail supplémentaire dans l'activité en amont pour s'accommoder aux changements impliqués par l'activité en aval. De manière à éviter les retours en arrière, on triangularise les DSM de manière à obtenir une séquence unidirectionnelle de flux d'information. En dernier recours, on choisit d'agréger les activités ou de les décomposer de manière à éliminer les retours en arrière. Ce processus nous permet d'obtenir une séquence d'exécution des activités, où les relations de précedence et les flux d'information sont orientés dans le même sens.

Dans la suite du papier, on suppose que des études préliminaires ont été menées de manière à identifier les relations entre les activités et à éliminer les retours en arrière. Les projets considérés sont donc composés uniquement d'activités dépendantes et indépendantes et le flux d'information (entre activités dépendantes) est considéré comme unidirectionnel.

La troisième étape du processus consiste à identifier les couples d'activités chevauchables. L'étude des échanges d'information entre les activités permet de différencier les couples d'activités en deux catégories : les couples chevauchables et les couples non chevauchables. Les premiers représentent les cas où l'activité en aval peut commencer avec des informations préliminaires et recevoir durant son exécution les informations finales de l'activité en amont. De ce fait, on peut chevaucher les deux activités, l'activité en aval peut commencer avant que l'activité en amont ne soit finie. Les activités non chevauchables sont connectées par des relations de précedence classiques fin-début, alors que les activités chevauchables sont connectées par des relations de type fin-début plus un retard, le retard étant le temps de chevauchement.

On autorisera les chevauchements lorsque la sensibilité, définie précédemment, sera faible. En effet, pour les couples d'activités dont la sensibilité est élevée, les changements dans l'information délivrée par l'activité en amont impliqueront de nombreuses retouches sur l'activité en aval. Le gain associé sera donc minime [Krishnan, 1996]. Dans la pratique, cela se résume à autoriser les chevauchements entre des activités d'ingénierie peu sensibles aux changements. A l'inverse, on évitera de chevaucher tous les couples d'activités comprenant des activités très sensibles aux changements.

Dans la suite du papier, nous noterons A et P l'ensemble des couples d'activités chevauchables et non chevauchables respectivement. De même nous noterons $A(i)$ et $P(i)$ l'ensemble

des prédécesseurs de l'activité i qui sont respectivement chevauchables et non chevauchables avec l'activité i .

$$Pred(i) = A(i) \cup P(i) \quad \forall i \in S$$

Les quatrième et cinquième étapes consistent à identifier les modes de chevauchement et les durées de retouche associés. La figure 2 représente le processus de chevauchement. L'activité j commence avec des informations préliminaires de l'activité i . Le taux de chevauchement associé au mode m , α_{ijm} , est exprimé comme une fraction de la durée de l'activité j . Une durée de retouche est par ailleurs ajoutée pour s'accommoder de la mise à jour des informations transmises lors de la réalisation de l'activité j . Le temps d'exécution des deux activités pour le mode de chevauchement m est donc égal à D_{ijm} exprimé comme suit :

$$D_{ijm} = d_i + d_j * (1 - \alpha_{ijm}) + r_{ijm}$$

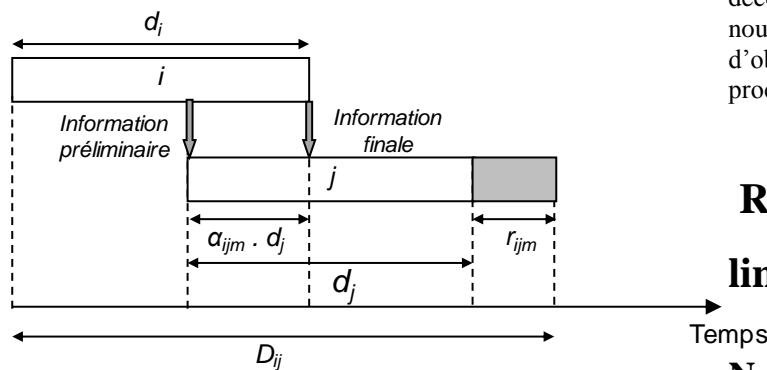


Figure 2 : Processus de chevauchement de deux activités i et j pour le mode m

Les pourcentages de chevauchement et les retouches associées aux modes sont définis de manière discrète. En effet, dans la réalité, l'ordonnancement d'un projet est établi sur la base de périodes discrètes (heures, jours, semaines, etc.). De plus, dans la pratique, l'avancement des activités est mesuré à partir de la réalisation de jalons qui correspondent à des avancements majeurs ou à la réalisation de biens livrables (spécifications, plan, commande de matière première, etc.) définis dans la phase initiale du projet. Les informations préliminaires nécessaires au commencement de l'activité en aval sont issues de la réalisation de ces jalons. Chaque mode de chevauchement est donc caractérisé par un pourcentage de chevauchement et une quantité de retouche supplémentaire à effectuer.

Les valeurs de α_{ijm} sont déterminées afin de correspondre avec la livraison des biens livrables majeurs de l'activité i . Cette méthodologie permet de s'assurer que les informations préliminaires utilisées sont des informations finalisées et approuvées. Le problème de la détermination du temps des retouches par rapport au temps de chevauchement est traité dans de nombreux articles de la littérature [Lin *et al.*, 2009.]. La définition des modes de chevauchement par rapport aux jalons permet de se baser facilement sur des données historiques.

Par la suite, les activités chevauchables peuvent être exécutées de manière conventionnelle sans chevauchement (mode 1), ou elles peuvent être chevauchées (mode > 1), voir figure 3. On suppose qu'il n'y a pas de restriction concernant le nombre de prédécesseurs chevauchables pour une activité. Dans ce cas, on considérera le total de retouches à effectuer comme la somme de retouches associées aux modes de précédences. Suite à l'évaluation des paramètres de chevauchement, nous pouvons faire l'ordonnancement du projet. Nous avons abordé deux techniques d'ordonnancement, l'une à l'aide d'un modèle de programmation linéaire, l'autre à l'aide de règle heuristique. Nous allons les présenter et comparer leurs résultats dans les sections suivantes.

Une fois l'ordonnancement terminé, on compare l'échéancier obtenu avec l'échéancier objectif, en termes de coût et de durée d'exécution. Si le temps d'exécution du projet ainsi que son coût associé sont acceptables, alors l'algorithme s'arrête ici. Le cas échéant, on recommence le processus à l'étape initiale, en décomposant plus finement les activités pour permettre de nouvelles opportunités de chevauchement qui permettront d'obtenir un échéancier plus compressé. On recommence le processus jusqu'à obtenir une solution acceptable.

Résolution par programmation linéaire

Notations

Un projet est défini par une liste d'activités S , comprenant deux activités fictives 0 et $n+1$ avec une durée d'exécution nulle, qui correspondent respectivement au début et à la fin du projet. On note d_j le temps d'exécution de l'activité j en considérant que toute l'information nécessaire des activités précédentes est disponible au début de l'activité j . Toutes les notations sont présentées dans le tableau 1. Les dates de fin au plus tôt et au plus tard correspondent en pratique aux dates désirées et ne correspondent pas forcément aux dates de fin au plus tôt et au plus tard calculées par une méthode d'ordonnancement.

Le modèle de minimisation du temps d'exécution du projet avec prise en compte du coût induit par le chevauchement

Le coût induit par le chevauchement est considéré comme l'addition des coûts supplémentaires de coordination induits par le chevauchement des activités, et des coûts supplémentaires de retouche. Dans notre papier, nous considérons la durée et le coût de coordination respectivement comme instantanée et négligeable. Le coût des retouches est à l'inverse égal au coût d'utilisation des ressources pour le travail supplémentaire. Nous supposons que les ressources impliquées dans les retouches sont les mêmes, en quantité et en qualification, que celles impliquées dans la réalisation de l'activité. Le coût total induit par la stratégie de chevauchement sera donc égal à :

$$C = \sum_{i \in A} \sum_{t=EF_i}^{LF_i} \sum_{m=1}^{m_i} \sum_{k \in R} X_{itm} * r_{im} * R_{ik} * \Theta_k \quad (1)$$

Le modèle de programmation linéaire est présenté ci-dessous :

$$\text{Minimiser } \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} + \delta * C \quad (2)$$

Tableau 1 : Définition des symboles

Symboles	Définition
S	Ensemble des activités
n	Nombre d'activités réelles
d_j	Durée d'exécution de l'activité j
A	Ensemble des couples d'activités chevauchables
P	Ensemble des couples d'activités non chevauchables
$A(j)$	Ensemble des prédécesseurs de j qui sont chevauchables avec l'activité j
$P(j)$	Ensemble des prédécesseurs de j qui ne sont pas chevauchables avec l'activité j
$Pred(j) = A(j) \cup P(j) \forall j \in S$	Ensemble des prédécesseurs de j
R	Ensemble des ressources
R_k	Quantité disponible constante pour la ressource k
R_{jk}	Nombre d'unités ressource k nécessaire par période pour l'exécution normale de l'activité j
Θ_k	Coût horaire de la ressource k
m_j	Nombre de mode d'exécution possible pour l'activité j
α_{ijm}	Pourcentage de la durée de chevauchement entre les activités i et j dans le mode d'exécution m , exprimé comme une fraction de d_j
r_{jm}	Durée des retouches associées au mode d'exécution m pour l'activité j
T	Durée maximale d'exécution du projet
$t = 0, \dots, T$	Périodes
EF_j	Date de fin au plus tôt de l'activité j
LF_j	Date de fin au plus tard de l'activité j

Contraintes :

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} (t - d_j \cdot (1 - \alpha_{ijm}) - r_{jm}) \cdot X_{jtm} \quad (3)$$

$\forall j \in S, \forall i \in Pred(j)$

$$\sum_{j=2}^n \left[R_{jk} \cdot \left(\sum_{m=1}^{m_j} \sum_{b=t}^{t+d_j-1+r_{jm}} X_{jbm} \right) \right] \leq R_k, \forall k \in R, \forall t \in [0, T] \quad (4)$$

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} t \cdot X_{jtm}, \forall i \in A(j) \quad (5)$$

$$\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} X_{jtm} = 1, \forall j \in S \quad (6)$$

$$0 \leq \delta * C < 1 \quad (7)$$

$$X_{jtm} = \{0,1\}, \forall j \in S, \forall t \in [0, T], \quad (8)$$

L'objectif (2) minimise la durée totale du projet tout en assurant une solution à coût minimale. La contrainte (3) prend en compte les différents modes de précedence applicables. La contrainte (4) définit les contraintes de ressources. La contrainte (5) permet de s'assurer que l'activité en aval ne puisse se terminer avant son prédécesseur, même en cas de chevauchement. En effet on suppose que l'activité en aval a besoin des informations finales du prédécesseur pour se terminer. La contrainte (6) s'assure qu'un seul mode de précedence est appliqué pour chaque activité et que chaque activité est terminée à la fin du projet. La contrainte (7) permet de prioriser la fonction objectif : on minimise d'abord le temps d'exécution du projet, pour lequel on recherche la solution à coût minimal. Enfin, la contrainte (8) définit la variable de décision binaire.

7.1 Cas d'étude

Nous avons utilisé le générateur de projet PROGEN développé par Kolish [Kolish et Sprecher, 1996] pour générer les projets. Les projets générés sont des projets de 30 activités plus deux activités fictives représentant le début et la fin du projet. Les relations de précedence ainsi que les durées d'activités sont générées par PROGEN. Les paramètres de chevauchement ($A, m_j, \alpha_{ijm}, r_{jm}$) sont générés ensuite. Nous rappelons que, pour les activités non chevauchables, un seul mode est réalisable et les taux de chevauchement et les quantités de retouches associées sont nuls. Les modes de chevauchement générés sont illustrés dans la figure 3. On génère pour chaque couple d'activités chevauchables, 4 modes de précedence, correspondant à des fractions de 0%, 25%, 50% et 75% de la durée d'exécution de l'activité en aval. Ce découpage est arbitraire et dans la pratique, les modes de précedences doivent correspondre aux différents jalons délivrés par l'activité en amont comme expliqué précédemment.

Pour notre cas d'étude, nous considérons que la quantité de retouche nécessaire est égale à 40% de la durée de chevauchement.

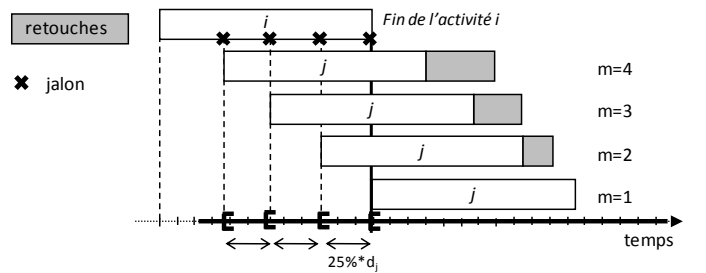


Figure 3 : Différents modes d'exécution possibles pour le couple (i,j)

Afin de mettre en lumière l'importance de la prise en compte des contraintes de ressources nous avons, pour chaque projet étudié, généré trois degrés de sévérité des contraintes de ressources grâce au paramètre RS défini par Kolisch *et al.* de la façon suivante :

$$R_k = Q_{k,\min} + \text{Round}(RS * (Q_{k,\max} - Q_{k,\min}))$$

$Q_{k,\min}$ est la quantité minimum acceptable pour la ressource k , égale à la demande maximale parmi les activités pour la ressource k . $Q_{k,\max}$ est égale à la demande maximale observée pour la ressource k lorsque l'on exécute le projet sans considérer les contraintes de ressources.

Les cas envisagés sont des contraintes sévères de ressources ($RS=0,5$), des contraintes moyennes ($RS=0,75$) et enfin l'absence de contrainte de ressources lorsque le projet est exécuté sans chevauchement ($RS=1$), où les ressources disponibles sont égales à la demande de ressource lors de l'exécution du projet sans chevauchement.

Résultats

Le modèle de programmation linéaire est résolu à l'aide d'AMPL Studio en utilisant le solveur CPLEX. La figure 4 illustre les différents gains obtenus suite à l'optimisation du planning de projet en prenant en compte les modes de chevauchement. On peut observer que plus les contraintes de ressources sont sévères, moins l'impact du chevauchement sur le temps d'exécution du projet est important. Pour le projet 3 par exemple, le gain varie de 15% pour des contraintes de ressources faibles à 4% pour des contraintes de ressources fortes. La figure illustre aussi l'augmentation du coût d'exécution du projet, associé au chevauchement d'activités. On observe que l'augmentation du coût n'est pas linéaire vis-à-vis de la réduction du temps d'exécution du projet, ce qui était prévisible du fait de l'aspect combinatoire du problème d'ordonnancement avec contrainte de ressources. Dans le tableau 2, on observe que les temps de résolution dépendent en grande partie de la sévérité des contraintes de ressources et du réseau d'activités considéré. Les temps de calculs observés sont trop importants et trop aléatoires pour que l'on puisse envisager une application du modèle à des projets industriels comprenant des centaines d'activités. En effet, le problème RCPSP est un problème NP difficile et on peut s'attendre à une augmentation significative des temps de calculs avec l'augmentation de la taille des problèmes. De plus, dans un contexte de re-planification, le temps de calcul se doit d'être limité au maximum afin de permettre une prise de décision rapide.

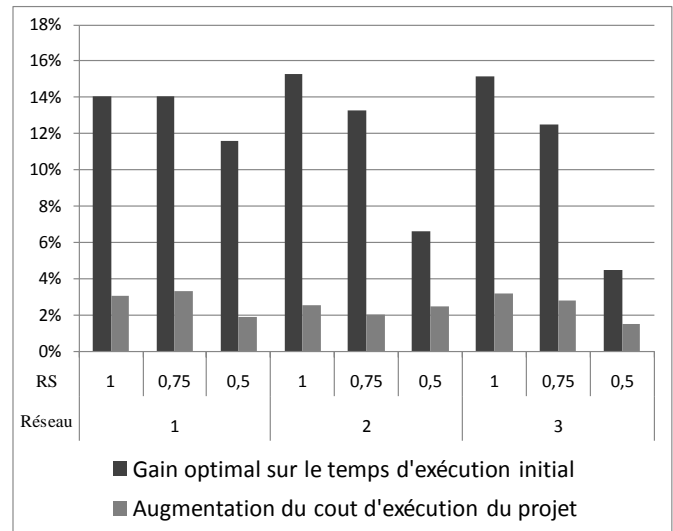


Figure 4 : Gain et coût associés à la résolution du modèle de programmation linéaire en nombre entier

Tableau 2 : Temps de calculs associés

1		2		3	
RS	Tps CPU (s)	RS	Tps CPU (s)	RS	Tps CPU (s)
1	9	1	5	1	3
0,75	28	0,75	13	0,75	7
0,5	93	0,5	>3600	0,5	21

A partir de ce constat, nous avons développé une approche heuristique qui vise à obtenir une bonne solution dans un temps de calcul raisonnable.

Résolution par approche heuristique

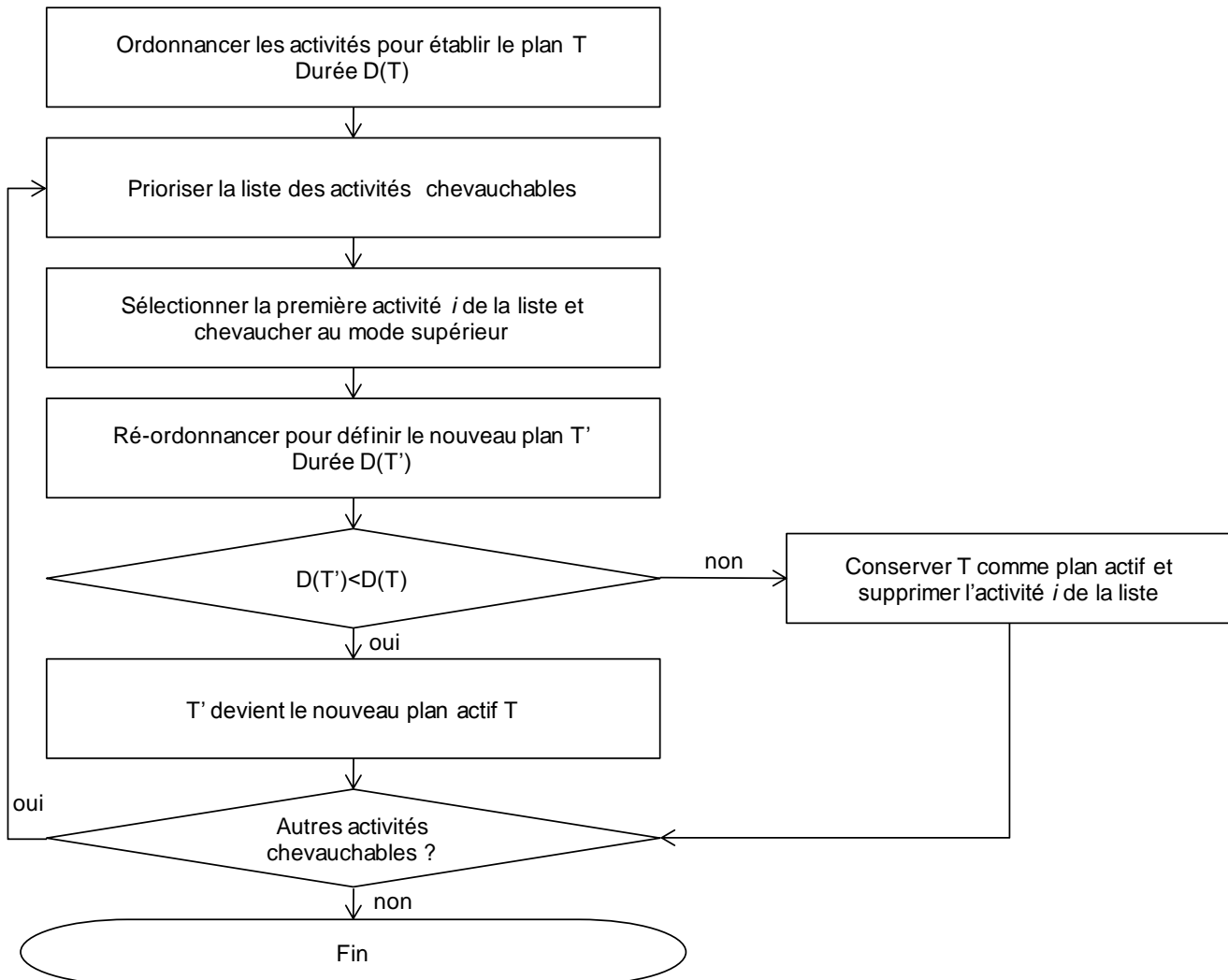
Présentation de l'algorithme

La figure 5 présente l'algorithme inspiré des méthodes de compression d'activité. La première étape consiste à ordonnancer le projet sans chevauchement d'activités afin d'obtenir un plan de référence. Ensuite on chevauche de manière incrémentale les activités selon un ordre de priorité. A chaque étape, on prend une décision par rapport au temps d'exécution du projet actif : si celui-ci est strictement inférieur au plan de référence, alors le plan actif devient le nouveau plan de référence. Sinon, on supprime l'activité de la liste des activités chevauchables. L'algorithme s'arrête lorsque qu'il n'y a plus d'opportunité de chevauchement dans la liste

Les activités chevauchables sont priorisées selon trois règles hiérarchisées. Tout d'abord, les activités sont classées en fonction de leur marge totale ; celles qui ont les marges totales les plus petites sont placées en premier. La marge totale est la différence des dates de fin d'une activité entre la planification au plus tard et la planification au plus tôt des activités. Les activités dont la marge totale est nulle sont les activités critiques du projet, elles ont une influence directe sur le temps d'exécution du projet. On ne peut cependant pas identifier de chemin critique du fait de la prise en compte des contraintes de ressources. En cas d'égalité de la marge totale entre plusieurs activités, on priorise les activités dont la quantité de ressource nécessaire à leur réalisation est la plus faible. Les activités mobilisant le moins de ressources sont en effet celles qui auront le moins d'impact sur

référence si le temps d'exécution actif est strictement inférieur au temps d'exécution de référence.

Nous avons implémenté l'algorithme dans un progiciel de gestion de projet, ici MS Project 2007. Ms Project est un progiciel de gestion de projet très répandu dans l'industrie. Les performances de son outil d'ordonnancement sont limitées [Kolisch, 1999] mais nous permettent de tester notre heuristique sur un outil utilisé dans la pratique. Les résultats obtenus et leurs comparaisons avec les résultats optimaux seront présentés dans la section suivante.



les contraintes de ressources. Enfin, en cas d'égalité, on classe les activités en fonction de leur nombre de successeurs, plus celui-ci est important, plus les activités seront mise en avant. La liste de priorité est actualisée et mise à jour à chaque itération.

On rappelle que la fonction d'évaluation de l'algorithme est la comparaison entre le temps d'exécution du plan actif et le temps d'exécution du plan de référence. Le plan actif devient le plan de

Résultats

Les temps de références, correspondant aux temps d'exécution du projet sans chevauchement d'activité, donné par MS Project ne sont pas les temps d'exécution optimaux. Pour chaque projet nous allons donc évaluer le gain par rapport au temps de référence donné par MS Project et non pas par rapport à l'optimal. La figure 6 illustre les différents gains et coûts associés aux projets suite à l'exécution de l'algorithme précédemment décrit.

On observe sur la figure 6 que les gains obtenus sur le temps d'exécution du projet sont moins importants en moyenne (10% au lieu de 12%) et plus aléatoires que les gains obtenus par la méthode précédente alors que les coûts sont comparables. Le nombre d'itérations nécessaires à l'obtention des résultats est inférieur à 150 pour l'ensemble des instances considérées ; chaque itération impliquant le calcul d'un ordonnancement du projet.

Cette première constatation nous permet de confirmer l'efficacité de l'algorithme pour accélérer l'exécution du projet et cela en un temps de calcul raisonnable.

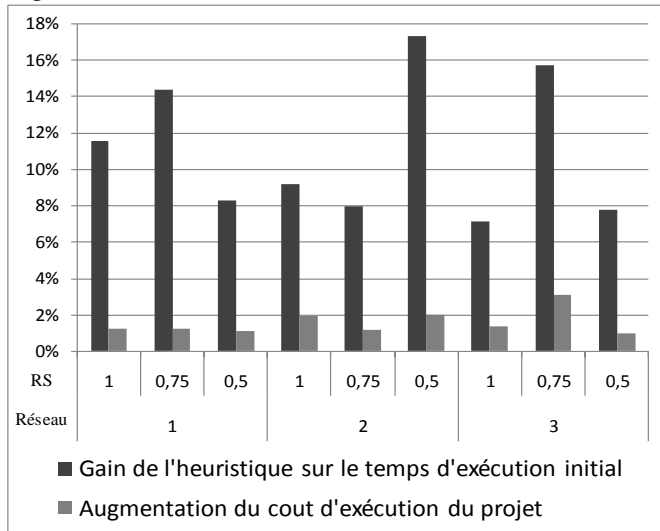


Figure 6 : Gain et coût associés à l'exécution de l'algorithme

Cependant la figure 6 ne nous permet pas d'évaluer la qualité des solutions obtenues. En effet, pour certaines instances on remarque que le gain sur le temps d'exécution peut être plus important avec l'heuristique, par exemple pour l'instance 2 avec RS=0,5, le gain observé est de 17% contre 7% à l'optimal. Cela s'explique par la valeur du temps de référence donnée par MS Project. Les gains observés précédemment ne sont pas comparables aux gains de la première méthode car ils ne se basent pas sur le même temps de référence.

Dans la figure 7, nous allons donc observer la qualité des solutions en les comparant aux solutions optimales. Une valeur de 0% signifie que la valeur trouvée par l'heuristique est égale à la valeur optimale. Une valeur supérieure à 0% illustre la marge d'erreur entre la valeur optimale et la valeur obtenue à la suite de l'exécution de l'algorithme.

La figure 7 illustre les marges d'erreur des temps de référence et des solutions données par l'algorithme. On observe tout d'abord que pour les temps de référence, l'erreur est d'autant plus importante que les contraintes de ressources sont sévères. Cela s'explique par la faible performance de l'outil d'ordonnancement utilisé (MS Project). La moyenne d'erreur pour les temps de référence est de 5% par rapport à l'optimal. Cette valeur est comparable avec les résultats donnés dans des études précédentes [Kolisch, 1999]. La qualité observée des solutions

suite à l'application de l'algorithme est variable, et la marge d'erreur moyenne par rapport à l'optimal est de 6%. En comparaison avec les temps de référence, il n'y a donc qu'un pourcent d'erreur supplémentaire induit par l'utilisation de l'algorithme pour l'obtention d'un échéancier accéléré prenant en compte les modes de chevauchement et les temps de retouche associés.

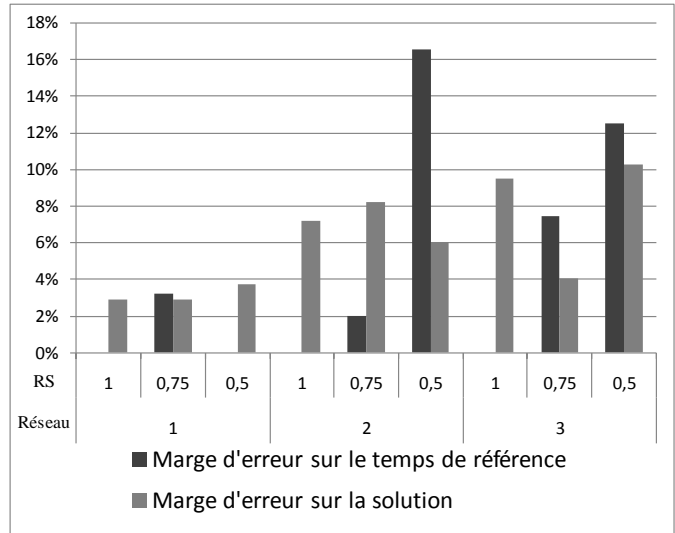


Figure 7 : Marges d'erreur associées à l'heuristique

On remarque aussi que l'utilisation de l'algorithme permet dans certains cas de compenser la faible performance de l'outil d'ordonnancement de projet, comme par exemple pour l'instance 2 avec RS=0,5. La marge d'erreur observée passe de 16% pour le temps de référence à 6% pour la solution. Aussi, dans nos résultats, l'utilisation de l'algorithme induit une erreur comprise entre 3% et 10% par rapport à l'optimal.

Le tableau 3 illustre le nombre d'itérations nécessaires pour atteindre la meilleure solution à l'aide de l'algorithme. Chaque itération correspond à un ordonnancement de projet. On observe que le nombre d'itération est assez faible et aléatoire. Aussi, l'application d'une heuristique nous permet de s'affranchir de la complexité du problème RCPSP multimode, le temps de résolution ne sera pas directement lié avec la taille de problème. L'application de l'heuristique nous permet donc d'obtenir une solution de bonne qualité dans un temps raisonnable.

Tableau 3 : Nombre d'itérations nécessaires à l'obtention de la solution

	Réseau	Nombre d'itération		
		1	2	3
RS	1	13	67	94
	0,75	14	5	135
	0,5	17	22	3

Nous avons aussi envisagé une autre règle de priorisation des activités. On classe les activités en fonction de leurs marges totales, de leurs nombres de successeurs et enfin de la quantité de ressources nécessaire à l'exécution normale de l'activité. Les

résultats obtenus sont similaires aux résultats obtenus précédemment, seule une instance donne un résultat supérieur au résultat précédent, les autres étant égales. Cependant cette seconde configuration a une influence sur le temps de résolution, augmentant en général le nombre d'itération nécessaires à l'obtention la solution. Cette remarque nous montre l'importance du choix des règles priorité, et l'importance d'envisager plusieurs règles dans des travaux futurs afin d'obtenir des résultats de meilleure qualité.

Conclusion

Le chevauchement d'activités est une des stratégies d'accélération de projet les plus utilisées, que ce soit lors de la planification initiale, ou lors d'une re-planification suite à des retards durant l'exécution du projet. Le chevauchement comprend des risques car l'activité en aval ne commence pas avec des informations confirmées et finalisées. Cependant, en pratique, les retouches nécessaires pour s'accommoder au changement dans l'information transmise par l'activité en amont ne sont pas toujours prises en compte. La littérature a beaucoup étudié la relation entre la durée de chevauchement et la durée de retouche nécessaire en fonction des caractéristiques d'un couple d'activités mais peu d'articles ont incorporé le chevauchement au problème d'ordonnancement. Cependant la résolution d'un problème de type RCPSP nécessite un temps de calcul conséquent pour des problèmes complexe, ce papier propose donc d'utiliser une méthode heuristique pour obtenir une solution de qualité acceptable en un temps réduit.

Nous avons joint le chevauchement d'activités à l'ordonnancement de projet avec les hypothèses suivantes : (1) les activités chevauchables sont identifiées préalablement, (2) les échanges d'information sont unidirectionnels, (3) les échanges d'information sont gratuits et instantanés, (4) les durées des chevauchements possibles sont définies par des modes (de manière discrète) et (5) les retouches sont estimées au préalable pour chaque mode de chevauchement. La contribution principale de ce papier est de présenter un processus d'accélération de projet et une heuristique permettant d'obtenir une solution de qualité acceptable en un temps réduit. L'heuristique développée s'inspire des méthodes de compression d'activités et a été implantée dans un progiciel de gestion de projet, MS Project. Nous avons aussi développé un modèle de programmation linéaire qui nous permet de comparer et d'évaluer la qualité des solutions obtenues.

Nos résultats montrent les limites d'un modèle de résolution linéaire pour une application industrielle et illustrent la pertinence d'une approche heuristique. Il serait intéressant par la suite de tester plusieurs règles de priorité pour l'algorithme afin de sélectionner la meilleure solution. Une implantation sur des projets réels de grande envergure permettrait d'évaluer avec plus de certitude l'applicabilité de notre processus et l'efficacité de l'heuristique proposée. Enfin, le développement de métaheuristiques permettrait de s'affranchir des problèmes de minimum local et d'obtenir ainsi des solutions de meilleure qualité.

Références

- Bogus S. M., Molenaar K. R., Diekmann J. E., (2005) Concurrent engineering approach to reducing design delivery time, *Journal of Construction Engineering and Management-Asce*, 131(11), pp. 1179-1185.
- Browning, T.R., (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions, *IEEE Transactions on engineering management*, 48(3), pp. 292-306.
- Gerk J. E. V., Qassim R. Y., (2008) Project Acceleration via Activity Crashing, Overlapping, and Substitution, *Ieee Transactions on Engineering Management*, 55(4), pp. 590-601.
- Kolisch, R., Sprecher, A., (1996) PSPLIB – A project scheduling problem library, *European Journal of Operational Research*, 96(1), pp. 205-216.
- Kolisch, R., (1999) Resource allocation capabilities of commercial project management software packages, *Interfaces*, 29(4), pp. 19-31.
- Krishnan V., Eppinger S. D., Whitney D. E., (1997) A model-based framework to overlap product development activities, *Management Science*, 43(4), pp. 437-451.
- Krishnan, V., (1996) Managing the simultaneous execution of coupled phases in concurrent product development, *IEEE Transactions on Engineering Management*, 43(2), pp. 210-217.
- Lin J., Chai K. H., Brombacher A. C., Wong Y. S., (2009) Optimal overlapping and functional interaction in product development, *European Journal of Operational Research*, 196(3), pp. 1158-1169.
- Maheswari J. U., Varghese K., (2005) Project Scheduling using Dependency Structure Matrix, *International Journal of Project Management*, 23(3), pp. 223-230.
- Steward D. V., (1981) The Design Structure System: A Method for Managing the Design of Complex Systems, *Ieee Transactions on Engineering Management*, EM-28(3), pp. 71-74.
- Wang, J. T., Lin, Y. I., (2009) An overlapping process model to assess schedule risk for new product development, *Computers & Industrial Engineering*, 57(2), pp. 460-474.

ANNEXE 3 : Programme “Accélération de projet avec mode de chevauchement”

‘Début de la procédure

Sub Accelration()

‘Définition des variables

Dim Njob As Integer

'Nombre d'activité

Dim Objectif As Integer

'Durée du plan optimal

Dim Objectif2 As Integer

'Durée du plan actuel

Dim Iteration As Integer

'Nombre d'itération maximal dans l'algorithme

Dim attente As Integer

'Temps en seconde nécessaire à l'actualisation des données dans

MsProject

Dim Gain As Integer

Dim Cout As Integer

Dim GainTotal As Integer

Dim GainTotalActive As Integer

Dim a() As Variant

'Pourcentage de chevauchement associé au couple (i,j) et au

mode m

Dim r() As Variant

'Durée de retouche pour l'activité i au mode m

ReDim Duree(1 To Njob)

ReDim ActiveMode(1 To Njob)

ReDim Relation(1 To Njob, 1 To 4)

ReDim ActiveModeTemp(1 To Njob)

compteur = 0

Limite = 100

'Temps d'attente pour la synchronisation des données

attente = Sheets("Resultat").Cells(4, 10).Value

'Nombre d'activité :

Njob = 32

'Nombre de mode maximum pour une activité, ici $4*4*4=64$:

MaxMode = 64

'Objectif2 est la durée du projet actuel dans MsProject :

Objectif2 = Sheets("Resultat").Cells(2, 3).Value

'Objectif est la durée minimale trouvée par l'heuristique, initialisé à Objectif2

Objectif = Objectif2

Iteration = 0

'Nombre de couple d'activité chevauché

NombreChevauchement = Sheets("Priorite").Cells(35, 2).Value

'Coût d'opportunité :

Alpha = 1000

'Acquisition des données

For k = 1 To Njob

Duree(k) = Sheets("DonneeInit").Cells(12 + k, 4).Value

Relation(k, 1) = Sheets("DonneeInit").Cells(12 + k, 18).Value

Relation(k, 2) = Sheets("DonneeInit").Cells(12 + k, 18 + 1).Value

Relation(k, 3) = Sheets("DonneeInit").Cells(12 + k, 18 + 2).Value

Relation(k, 4) = Relation(k, 1)

ActiveMode(k) = 1

If Relation(k, 2) > 0 Then Relation(k, 4) = Relation(k, 4) & ";" & Relation(k, 2)

If Relation(k, 3) > 0 Then Relation(k, 4) = Relation(k, 4) & ";" & Relation(k, 3)

Next k

ReDim a(1 To Njob, 1 To Njob, 1 To MaxMode)

ReDim r(1 To Njob, 1 To MaxMode)

For j = 1 To Njob

For i = 1 To Njob

For m = 1 To MaxMode

a(j, i, m) = 0

r(j, m) = 0

Next m

Next i

Next j

k = 2

Do While Sheets("Chevauchement").Cells(k, 5).Value <> 0

j = Sheets("Chevauchement").Cells(k, 5).Value

i = Sheets("Chevauchement").Cells(k, 5 + 1).Value

m = Sheets("Chevauchement").Cells(k, 5 + 2).Value

a(j, i, m) = Sheets("Chevauchement").Cells(k, 5 + 4).Value

k = k + 1

Loop

k = 2

Do While Sheets("Chevauchement").Cells(k, 11).Value <> 0

j = Sheets("Chevauchement").Cells(k, 11).Value

m = Sheets("Chevauchement").Cells(k, 11 + 1).Value

r(j, m) = Sheets("Chevauchement").Cells(k, 11 + 2).Value

k = k + 1

Loop

'Initialisation des modes de chevauchement

```

valeurtest = 1
Iteration = 0
l = 0
Do While Sheets("Resultat").Cells(l + 7, 2).Value <> 0
    ActiveMode(Sheets("Resultat").Cells(l + 7, 2).Value) = Sheets("Resultat").Cells(l + 7, 3).Value
    l = l + 1
Loop

For k = 1 To Njob
    If ActiveMode(k) > 1 Then
        For l = 1 To 3
            If Relation(k, l) > 0 Then
                test = 1
                Sheets("Donnee").Cells(12 + k, 9 + l).Value = Relation(k, l) & "FD-" & a(k, Relation(k, l),
ActiveMode(k)) & " jours"
                Call tempo(attente)
            End If
        Next l
        Sheets("Donnee").Cells(12 + k, 4).Value = Duree(k) + r(k, ActiveMode(k))
        Call tempo(attente)
    End If
Next k

```

‘Algorithme d’accélération

'Actualisation des données de chevauchement

```

Do While compteur <= Limite
    If valeurtest = 0 Then compteur = Limite + 1
    If (valeurtest = 1) Then
        Tri
        Sheets("Resultat").Select
        Call ecriture(Njob, ActiveMode, Objectif, Iteration, debut)
        k = 0
        valeurtest = 0
    End If

```

Do While k < NombreChevauchement

'Choix de l'activité dans la liste

ActiviteChev = Sheets("Priorite").Cells(2 + k, 1).Value

'Mise à jour des modes de précédence, pour l'activité sélectionnée on choisi le mode supérieur

```

For p = 1 To Njob
    ActiveModeTemp(p) = ActiveMode(p)
Next p
Mode = ActiveMode(ActiviteChev) + 1

```

ActiveModeTemp(ActiviteChev) = Mode
 valeurcontrole = 0

Do While Mode < MaxMode

'Actualisation des valeurs de chevauchement

ActiveModeTemp(ActiviteChev) = Mode

valeurchevauchement = 0

For l = 1 To 3

If Relation(ActiviteChev, l) > 0 Then

valeurchevauchement = valeurchevauchement + a(ActiviteChev, Relation(ActiviteChev, l), Mode)

End If

Next l

If valeurchevauchement > 0 Then

Sheets("Donnee").Cells(12 + ActiviteChev, 4).Value = Duree(ActiviteChev) + r(ActiviteChev, Mode)

Call tempo(attente)

test = 0

For l = 1 To 3

If Relation(ActiviteChev, l) > 0 Then

test = 1

Sheets("Donnee").Cells(12 + ActiviteChev, 9 + l).Value = Relation(ActiviteChev, l) & "FD-" & a(ActiviteChev, Relation(ActiviteChev, l), Mode) & " jours"

Call tempo(attente)

End If

Next l

Iteration = Iteration + 1

Call tempo(attente)

‘*****

'Evaluation de la solution

'Mise à jour de Objectif2 et de la valeur objectif (ici on présente la fonction d'évaluation qui permet d'obtenir le meilleur compromis entre diminution du temps d'exécution et augmentation des coûts, scénario 2)

Objectif2 = Sheets("Resultat").Cells(2, 3).Value

GainTemps = Objectif2 - Objectif

Cout = CalculCout(Njob, r, ActiveModeTemp)

GainTotal = Beta * Cout + Alpha * GainTemps

'En cas d'égalité on annule la décision de chevauchement qui a été reprise et on essaye avec le mode supérieur

If GainTotal = GainTotalActive Then

'If Objectif2=Objectif then... pour le scenario 1

Sheets("Donnee").Cells(12 + ActiviteChev, 4).Value = Duree(ActiviteChev) + r(ActiviteChev, ActiveMode(ActiviteChev))

For l = 1 To 3

If Relation(ActiviteChev, l) > 0 Then

test = 1


```

    Sheets("Donnee").Cells(12 + ActiviteChev, 9 + 1).Value = Relation(ActiviteChev, 1) & "FD-" &
a(ActiviteChev, Relation(ActiviteChev, 1), ActiveMode(ActiviteChev)) & " jours"
    End If
    Next l
    Call tempo(attente)
    Mode = Mode + 1
    valeurcontrole = valeurcontrole + 1
    If valeurcontrole >= 3 Then Mode = MaxMode + 1
End If

```

'Si le gain est inférieur, alors on annule la décision et on choisit l'activité suivante dans la liste

```

    If GainTotal < GainTotalActive Then
'If Objectif2>Objectif then ... pour le scenario 1
        GainTotalActive = GainTotal
        ActiveMode(ActiviteChev) = Mode
        Mode = MaxMode + 1
        k = NombreChevauchement + 1
        valeurtest = 1
    End If

```

'Si le gain est supérieur, alors le mode actuel devient le mode actif pour l'activité chevauché

'On effectue alors a nouveau un tri et on sélectionnera la première activité

```

    If GainTotal > GainTotalActive Then
'If Objectif2>Objectif then ... pour le scenario 1
        Sheets("Donnee").Cells(12 + ActiviteChev, 4).Value = Duree(ActiviteChev) + r(ActiviteChev,
ActiveMode(ActiviteChev))
        Call tempo(attente)
        For l = 1 To 3
            If Relation(ActiviteChev, l) > 0 Then
                Sheets("Donnee").Cells(12 + ActiviteChev, 9 + 1).Value = Relation(ActiviteChev, l) & "FD-" &
a(ActiviteChev, Relation(ActiviteChev, l), ActiveMode(ActiviteChev)) & " jours"
                Call tempo(attente)
            End If
        Next l
        Mode = MaxMode + 1
    End If

```

Else:

```

    Mode = Mode + 1
    valeurcontrole = valeurcontrole + 1
    If valeurcontrole >= 3 Then Mode = MaxMode + 1
End If

```

```

    Call tempo(attente)
    Call ecriture(Njob, ActiveMode, Objectif, Iteration, debut)
    Sheets("Resultat").Cells(8, 6).Value = GainTotalActive
    Sheets("Resultat").Cells(8, 7).Value = CalculCout(Njob, r, ActiveMode)
    Loop
    k = k + 1

```

```

    Loop
compteur = compteur + 1
Loop

```

'On écrit tous les choix de chevauchement qui ont été pris

```

For k = 1 To Njob
Sheets("Donnee").Cells(12 + k, 4).Value = Duree(k) + r(k, ActiveMode(k))
For l = 1 To 3
If Relation(k, l) > 0 Then
If a(k, Relation(k, l), ActiveMode(k)) > 0 Then
    Sheets("Donnee").Cells(12 + k, 9 + l).Value = Relation(k, l) & "FD-" & a(k, Relation(k, l), ActiveMode(k)) &
    " jours"
End If
End If
Next l
Next k
Sheets("Resultat").Cells(8, 6).Value = GainTotalActive
Sheets("Resultat").Cells(8, 7).Value = CalculCout(Njob, r, ActiveMode)

Call ecriture(Njob, ActiveMode, Objectif, Iteration, debut)

```

End Sub

'Fin de la procédure